



Escuela
Politécnica
Superior

Implementación de un sitio web de consulta de literatura valenciana



Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor:

Javier Monllor Alcaraz

Tutor/es:

Jaume Aragonés Ferrero



Universitat d'Alacant
Universidad de Alicante

Enero 2020

Agradecimientos

Primero quisiera agradecer a mis supervisores Jaume Aragonés Ferrero y Gabriel Sansano i Belso. Permitieron constantemente que este trabajo me resultara súper interesante y que en ningún momento tirase la toalla, me guiaron en la dirección correcta cada vez que pensaban que lo necesitaba.

También estoy muy agradecido a mis amigos y amigas, compartieron conmigo estos cuatro años de carrera haciendo que todo este proceso fuera mucho más llevadero. En especial a Álvaro Girones García y Manuel Morote Herrero que he aprendido mucho de ellos dos y siempre hemos estado juntos hasta el final.

Finalmente, quiero expresar mi profunda gratitud a mis padres por brindarme un apoyo incondicional y aliento continuo durante mis años de estudio. Este logro no hubiera sido posible sin ellos. Gracias.

Contenido

1	INTRODUCCIÓN	1
1.1	MOTIVACIÓN.....	1
1.2	OBJETIVOS.....	2
2	TECNOLOGÍAS ACTUALES DE CREACIÓN DE SITIOS WEB	4
3	TECNOLOGÍAS EMPLEADAS EN EL PROYECTO	9
3.1	SOFTWARE.....	9
3.1.1	PHP.....	10
3.1.2	Laravel.....	11
3.1.3	Laravue.....	12
3.1.4	Vue.....	14
3.1.5	MySQL.....	15
3.1.6	Git.....	16
3.1.7	Bullzip.....	18
4	METODOLOGÍA	21
4.1	¿QUÉ ES LA METODOLOGÍA ÁGIL?.....	21
4.2	¿CÓMO LAS HE UTILIZADO?.....	22
4.2.1	<i>Scrum</i>	22
4.2.2	<i>Kanban</i>	23
4.3	¿POR QUÉ LAS HE UTILIZADO?.....	25

5	ANÁLISIS	27
5.1	CONVERTIR LA BASE DE DATOS DE ACCESS A MYSQL.....	27
5.2	CAMBIOS EN LA BASE DE DATOS.....	28
5.3	DISEÑAR LOS DIAGRAMAS DE ER.....	33
5.4	IMPLEMENTAR LOS <i>MOCKUPS</i>	38
5.5	CREAR EL REPOSITORIO DE GITHUB Y EL PROYECTO DE LARAVEL...	48
6	IMPLEMENTACIÓN	50
6.1	IMPLEMENTAR EL <i>FRONT OFFICE</i>	50
6.1.1	Errores.....	69
6.2	RESULTADO DEL PROYECTO VS <i>MOCKUPS</i>	70
6.3	IMPLEMENTAR EL <i>BACK OFFICE</i>	79
7	PUESTA EN PRODUCCIÓN	97
8	PRUEBAS DEL SISTEMA	100
8.1	PRUEBAS UNITARIAS.....	100
8.1.1	<i>Fron-end</i>	101
8.2	PRUEBAS DE FUNCIONALIDAD.....	101
8.2.1	<i>Front-end</i>	101
8.2.2	<i>Back-end</i>	102
8.3	PRUEBAS DE INTERFAZ/USABILIDAD.....	103
8.3.1	<i>Front-end</i>	103
8.3.2	<i>Back-end</i>	104

9	PROBLEMAS Y SOLUCIONES	106
9.1	BASE DE DATOS.....	106
9.1.1	Solución.....	107
9.2	INTEGRACIÓN DE LARAVUE.....	113
9.2.1	Solución.....	113
9.3	DATOS QUE NO RELACIONAN A NINGUNA TABLA.....	114
9.3.1	Solución.....	115
10	CONCLUSIÓN Y TRABAJOS FUTUROS	118
10.1	FUNCIONALIDADES NO IMPLEMENTADAS.....	119
10.2	AMPLIACIONES DE FUNCIONALIDAD.....	120
10.3	PRUEBAS POR REALIZAR.....	121
11	REFERENCIAS	123

Listado de Figuras

4.1	Ejemplo de listado de issues de Github.....	23
4.2	Tablero Kanban del proyecto creado en Github.....	24
5.1	Diagrama de Eautors -> donde la tabla principal es la los autores.....	34
5.2	Diagrama de Eobres -> donde la tabla principal es la de obras.....	35
5.3	Diagrama de Epublicacio -> donde la tabla principal es la de publicaciones.....	36
5.4	Diagrama de MM -> donde aparecen todas las relaciones de M:M.....	37
5.5	Página principal con el primer tab.....	39
5.6	Página principal con el segundo tab.....	40
5.7	Página principal con el tercer tab.....	41
5.8	Página principal con el cuarto tab.....	42
5.9	Página de información del autor.....	43
5.10	Página de perfil del autor.....	44
5.11	Página de información de una obra.....	45
5.12	Página de información de un idioma.....	46
5.13	Página de información de un premio.....	47
6.1	Fichero tabs donde se carga el contenido del abecedario dependiendo del tab seleccionado.....	63
6.2	Fichero abecedari con bucles for para construir los botones del abecedario.....	64
6.3	Resultado del código de la Figura 6.1 y Figura 6.2.....	65

6.4	Función Javascript que comprueba el origen del usuario, localiza el componente select y la función a invocar.....	66
6.5	Función AJAX que invoca la función localizada en la Figura 6.4 e inyecta el resultado en el componente select localizado en la Figura 6.4.....	67
6.6	Función que devuelve todos los autores cuya letra del primer apellido empiece por la letra pasado por parámetro.....	68
6.7	Resultado de aplicar el código de la Figura 6.4, Figura 6.5 y Figura 6.6.....	69
6.8	El usuario busca sin seleccionar una letra o autor/a.....	70
6.9	Página principal con el primer tab: Índex Autors/es.....	71
6.10	Página principal con el segundo tab: Cercador Autors/es.....	72
6.11	Página principal con el tercer tab: Obres.....	72
6.12	Página principal con el cuarto tab: Cercador Obres.....	73
6.13	Página principal con el quinto tab: Altres Consultes.....	74
6.14	Página de información del autor.....	75
6.15	Página de información de una obra.....	76
6.16	Página de información de un premio.....	77
6.17	Página de información de un idioma.....	77
6.18	Footer.....	78
6.19	Aspecto de back office inicial.....	80
6.20	Aspecto del back office final.....	81
6.21	Listado de todos los autores.....	82
6.22	Añadir un nuevo autor al pulsar el botón Crear.....	84
6.23	Modificar un autor al pulsar el botón Modificar.....	86
6.24	Eliminar un autor al pulsar sobre el botón Eliminar.....	87
6.25	Listado de todas las obras.....	88
6.26	Añadir una nueva obra al pulsar el botón Crear.....	89
6.27	Modificar una obra al pulsar el botón Modificar.....	90

6.28	Eliminar una obra al pulsar sobre el botón Eliminar.....	91
6.29	Listado de todas las publicaciones.....	91
6.30	Añadir una nueva publicación al pulsar el botón Crear.....	93
6.31	Modificar una publicación al pulsar el botón Modificar.....	94
6.32	Eliminar una publicación al pulsar sobre el botón Eliminar.....	95
7.1	Esquema del proyecto.....	98
9.1	Localizar la base de datos a convertir e identificación si es necesario.....	108
9.2	Nombre de la nueva base de datos y localización del fichero.....	109
9.3	Tablas a convertir.....	110
9.4	Configuraciones de la nueva base de datos.....	111
9.5	Estructuración de carpetas en Github.....	113
9.6	Vista para el usuario cuando hay errores en las relaciones de la base de datos.....	115

Listado de Tablas

5.1	Nuevos campos en las tablas.....	28
5.2	Campos eliminados.....	29
5.3	Nuevas tablas con sus campos y relaciones.....	30
5.4	Tablas eliminadas.....	32
6.1	Información del controller AltresAutorsController.....	52
6.2	Información del controller CapsaleraController.....	52
6.3	Información del controller ConsultaAltresController.....	53
6.4	Información del controller ConsultaObraController.....	54
6.5	Información del controller ContactarController.....	55
6.6	Información del controller EditorialController.....	56
6.7	Información del controller IlustradorController.....	56
6.8	Información del controller IndexController.....	57
6.9	Información del controller PremiController.....	58
6.10	Información del controller ResultatAutorController.....	58
6.11	Información del controller TabsController.....	59
6.12	Rutas principales del sitio web.....	60

Capítulo 1

Introducción

Este documento presenta el trabajo de final de grado en Ingeniería Informática de la Universidad de Alicante centrado en la implementación de un sitio web de consulta de literatura valenciana actual. El proyecto fue dirigido por el profesor Jaume Aragonés Ferrero y supervisado por el profesor Gabriel Sansano i Belso, el cual fue el director del proyecto de investigación en el que se basa el sitio web.

1.1 Motivación

La motivación principal ha sido poder recuperar una web que hacía mucho tiempo no funcionaba porque la tecnología aplicada en su momento se encuentra en estado obsoleto. La web tenía bastantes visitas por parte de centros educativos y era conocida entre los propios escritores los cuales enviaban y siguen enviando sus datos y actualizaciones para que sean reflejadas en el sitio web. El poder revivir esta web a mi gusto y diseño ha sido una de las mayores motivaciones sin duda a lo largo de la carrera.

1.2 Objetivos

El objetivo principal de este proyecto es modernizar la antigua página de literatura valenciana actual, implementada por mi tutor, y hacerla funcional con las nuevas tecnologías impuestas a día de hoy. Se ha de implementar un sitio web de consulta de autores y obras de escritores valencianos modernos (a partir de un determinado año), además de un acceso para la gestión de los datos (*back office*) para que así el administrador pueda realizar los cambios correspondientes. En este caso, la web será administrada por el profesor Biel o alguien de su grupo de investigación, ya que él es quien le pidió a mi tutor implementar la web en el pasado.

Capítulo 2

Tecnologías actuales de creación de sitios web

Hace muchos años, las aplicaciones se implementaban para funcionar en entorno de escritorio, por lo que se debía realizar una instalación y mantenimiento *in situ*. Cada vez que se obtenía una nueva versión, el cliente debía de realizar una instalación del programa para poder observar los nuevos cambios, cosa que hoy en día resulta impensable. Con la mejora de las redes y la implantación de los servicios *on-line*, se prefiere optar por desarrollar en entorno web (cliente/servidor) y de esta forma se centraliza la instalación y actualización del software. Es por ello que actualmente, al subir una nueva versión a producción, el cliente la puede observar desde su puesto de trabajo, su ordenador personal o móvil.

Otra cosa muy importante sobre las aplicaciones web, es que permiten que se pueda acceder tanto por móvil, Tablet o PC. En conclusión, las empresas hoy en día se vuelcan en realizar aplicaciones en la web (*intranets*, *cloud computing*) más que de escritorio debido a:

- Ahorro de costes de hardware y software -> Sólo es necesario un ordenador con navegador y Internet.
- Uso fácil.
- Uso simultáneo, accesibles desde cualquier lugar.
- Escalables y rápida actualización.
- Menos propensas a colgarse y crear problemas.

- Datos más seguros.

Para desarrollar un sitio web hay múltiples opciones tecnológicas, tanto de cliente como de servidor:

Cliente:

- HTML, CSS, Javascript
- [4] Bootstrap
- *Frameworks de front-end:*
 - Angular
 - Vue.js
 - React
 - Redux

Servidor:

- Java
- ASP.net MVC, Core
- PHP
- Node.js
- Python
- *Frameworks de back-end:*
 - Spring
 - Play Framework
 - Laravel
 - Laravue

La decisión sobre qué tecnología usar viene determinada por:

- Facilidad en el lenguaje:
 - Tener alguna experiencia en el lenguaje.
 - Gran cantidad de librerías y marcos.
 - Estructurado y conciso.
 - Sintaxis no muy compleja.
- Popularidad:
 - Preparación de cara al mundo profesional.
- Versatilidad:
 - Adaptarse rápido y fácil a diferentes funciones.
- Software de código abierto.
- Plataforma de uso:
 - Windows:
 - Su alta compatibilidad lo convierte en el entorno de desarrollo adecuado.
 - MacOS:
 - Interactúa con los compiladores de Swift, Objective-C y C / C++.
 - Linux:
 - Programación más intensa.
 - Alto nivel de personalización.
- Implementación de *front-end* y *back-end*:
 - *Front-end*:
 - Javascript.
 - *Back-end*:
 - Python, Java, C, C++ o PHP.
- Propósito de la programación:
 - Centrado en la empresa:
 - Invertir tiempo y dinero.

- Centrado en el uso personal:
 - Lenguaje gratis, abierto y fácil de aprender.

Por todo lo anteriormente dicho, la elección de las tecnologías para implementar el sitio web no fue muy complicada. Por un lado, para la implementación del *front office* he utilizado el lenguaje PHP, ya que aparte que cumple con varios puntos comentados anteriormente, se basa en Laravel que es el *framework* utilizado. Además, he utilizado Bootstrap (*framework* CSS) para crear una interfaz de usuario limpia y totalmente adaptable a todo tipo de dispositivos y pantallas. Por otro lado, para la implementación del *back office* he decidido utilizar Laravue, ya que al tratarse de una mezcla entre Laravel y Vue.js, considero que hoy en día el *framework* Vue.js tiene muchas salidas de cara al mundo laboral y durante la carrera apenas he sabido nada de él, pienso que podría suponerme un reto mayor el poder aprender este *framework* por mi cuenta y a mi gusto. En el siguiente apartado, comentaré con más detalle qué tecnologías he utilizado, cómo las he utilizado y por qué las he utilizado.

Capítulo 3

Tecnologías empleadas en el proyecto

La elección de las tecnologías que se utilizarían para llevar a cabo el desarrollo del proyecto fue uno de los objetivos más importantes en las primeras fases de este proyecto. En este capítulo, describiré las herramientas que se emplean, directa o indirectamente, en el desarrollo de este proyecto, revelando las principales motivaciones que me han llevado a elegir las.

3.1 Software

La parte del *front office* ha sido implementada utilizando el lenguaje PHP con la ayuda del Framework Laravel junto con sus librerías, mientras que la parte del *back office* ha sido implementada utilizando Laravue. He utilizado Git, un sistema de control de versiones de código abierto, para gestionar el proyecto. El repositorio se encuentra alojado en Github, en mi caso se trata de un repositorio privado el cual solo he permitido acceso a mi tutor. El código se ha escrito utilizando diferentes editores de texto, entre los cuales Visual Studio Code se destaca por sus capacidades de autocompletado, al igual que Sublime Text por su capacidad de buscar campos de una forma rápida y eficiente. Se han utilizado dos herramientas en línea para crear los *mockups* y diagramas:

- [6] Sitio web online de creación de *mockups*
- <https://www.draw.io> (arquitecturas siamesas)

Por último, se ha utilizado un programa para poder convertir la base de datos inicial del antiguo proyecto “Literatura valenciana actual” de Access a MySQL. El programa utilizado ha sido Bullzip MS Access to MySQL.

A continuación, voy a describir las diferentes tecnologías software que he utilizado a lo largo de la implementación de ambos proyectos.

3.1.1 PHP

- ¿Qué es?



PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas. PHP se utiliza para generar páginas web dinámicas.

- ¿Cómo lo he utilizado?

Para implementar el proyecto, he utilizado la versión PHP 7.2.12 ya que es compatible con la versión del Framework Laravel que he usado.

- ¿Por qué lo he utilizado?

Claramente tenía que utilizar el lenguaje PHP porque Laravel, el *framework* del proyecto se basa en PHP. Además, las cosas buenas que tiene PHP es que se tiene la libertad de escoger el sistema operativo y el servidor web, tiene la posibilidad de utilizar programación por procedimientos o programación orientada a objetos, o una mezcla de ambas. Por último, una de las características más potentes y destacables es su soporte para un amplio abanico de bases de datos.

3.1.2 Laravel

- ¿Qué es?



Laravel es uno de los *frameworks* de código abierto más fáciles de asimilar para PHP. Es simple, muy potente y tiene una interfaz elegante y divertida de usar. Fue creado en 2011 y tiene una gran influencia de *frameworks* como Ruby on Rails, Symfony y ASP.NET MVC. El objetivo de Laravel es el de ser un *framework* que permita el uso de una sintaxis refinada y expresiva para crear código de forma sencilla, evitando el «código espagueti» y permitiendo multitud de funcionalidades. Aprovecha todo lo bueno de otros *frameworks* y utiliza las características de las últimas versiones de PHP. La mayor parte de su estructura está formada por dependencias, especialmente de Symfony, lo que implica que el desarrollo de Laravel dependa también del desarrollo de sus dependencias.

- ¿Cómo lo he utilizado?

Para implementar el proyecto, he utilizado la versión 5.5 de Laravel ya que es la que he utilizado a lo largo de toda la carrera y con la que más manejo tengo.

- ¿Por qué lo he utilizado?

En primer lugar, comentar que he decidido utilizar Laravel para implementar la parte de *front office* ya que a lo largo de la carrera me he sentido muy cómodo con él, me pareció que el esquema de MVC se veía perfectamente cómo se separaba cada parte. Mi motivación para haberlo escogido es que quería especializarme y aprender nuevas cosas de Laravel, las cuales durante la carrera no dio tiempo a profundizar.

Una motivación extra fue el hecho de que hoy en día, el *framework* de Laravel es muy utilizado por muchas empresas, por ello mi gran objetivo de cara al mundo laboral era aprender lo máximo posible para en un futuro poder tener mayores oportunidades.

3.1.3 Laravue

- ¿Qué es?



laravue

Laravue surge de la combinación de un proyecto de Laravel con la utilización de Vue.js en el *front-end*. De esta forma, al utilizar esta plantilla del proyecto, es posible implementar un *back office* de forma rápida y cómoda,

puesto que Laravue nos provee de diversas utilidades y herramientas para tener listo en poco tiempo un mantenimiento (*CRUD*) de las entidades de nuestro sistema de información.

- ¿Cómo lo he utilizado?

[7] Investigando, descubrí que existe un proyecto base implementado en Laravue el cual creí que me podía servir como punto de partida para implementar la parte del *back office*. Tras clonar el repositorio, me di cuenta que ese proyecto era totalmente distinto al que yo había implementado en Laravel y por lo tanto se tenía que subir a Github como otro proyecto distinto.

- ¿Por qué lo he utilizado?

Mi tutor me aconsejó utilizar Laravue para implementar la parte del *back office*. No sabía realmente lo que era, pero investigando descubrí que era una mezcla de Laravel + Vue.js. Al igual que en el caso anterior, mi motivación principal para escoger esta opción y no otra es el hecho de que Vue.js es un *framework* de Javascript que hoy en día, poco a poco se está convirtiendo en uno de los *frameworks* que representa un mayor uso en el mercado. Este *framework*, durante la carrera lo he estudiado y utilizado bastante poco en la asignatura de Aplicaciones Distribuidas en Internet. En mi opinión, me quedé con ganas de saber y conocer más sobre este *framework*, por ello durante la implementación de mi proyecto me ha supuesto un reto extra el poder aprender nuevas funcionalidades y usos de este *framework*, que por mi parte me he quedado satisfecho de mi trabajo.

3.1.4 Vue

- ¿Qué es?



Vue es un *framework open source* de JavaScript, el cual nos permite construir interfaces de usuarios de una forma muy sencilla. La curva de aprendizaje es relativamente baja, debes conocer muy bien JavaScript, saber trabajar con *callbacks*, promesas, objetos, entre otros temas más. Una de las características más importantes de Vue es el trabajo con componentes. Un componente Vue, en términos simples, es un elemento el cual se encapsula código reutilizable. Dentro de un componente podremos encontrar etiquetas HTML, estilos de CSS y código JavaScript. Los componentes nos permiten desarrollar proyectos modularizados y fáciles de escalar, si nosotros así lo deseamos podemos reemplazar un componente por otro de una forma muy sencilla, como si de piezas de lego se trataran.

- ¿Cómo lo he utilizado?

En mi caso, Vue lo uso en el proyecto de Laravue comentado anteriormente para construir el *back office*.

- ¿Por qué lo he utilizado?

Al igual que en el caso de PHP, he tenido que utilizar Vue ya que Laravue surge de la suma: Laravel + Vue, por tanto no podía utilizar otro Framework de Javascript. Además tenía la motivación extra que he comentado anteriormente sobre el *Framework* que quería investigar por mi cuenta para

aprender muchas más cosas de las que ya se, pensando de cara al mundo laboral y a mi formación profesional.

3.1.5 MySQL

- ¿Qué es?



MySQL es un sistema de gestión de bases de datos que cuenta con una doble licencia. Por una parte es de código abierto, pero por otra, cuenta con una versión comercial gestionada por la compañía Oracle. Actualmente, es la base de datos de código abierto más famosa y utilizada en el mundo entero. Una de las principales características de MySQL es que trabaja con bases de datos relacionales, es decir, utiliza tablas múltiples que se interconectan entre sí para almacenar la información y organizarla correctamente.

- ¿Cómo lo he utilizado?

Como versión de MySQL he utilizado la versión 5.7.23, luego he utilizado como herramienta visual de diseño de bases de datos MySQL Workbench 8.0 CE, el cual lo he utilizado para realizar las siguientes tareas:

- Creación de nuevas tablas o campos
- Modificación de campos
- Borrado de tablas o campos
- Insertar nuevos datos para hacer pruebas
- Modificar datos para hacer pruebas
- Realizar consultas

- ¿Por qué lo he utilizado?

He decidido utilizar como gestor de bases de datos MySQL, aparte de ser actualmente uno de los más utilizados y reconocidos en el mercado, es el que he utilizado y aprendido durante toda la carrera. Es mayormente utilizado en conjunción con servidores web donde se lo encuentra relacionado a aplicaciones web o CMS para sitios online y está muy ligado a PHP en lo que se refiere a este tipo de desarrollos. Además para mi proyecto en Laravel, es muy sencillo conectar la base de datos, basta con modificar el archivo `.env` con las credenciales establecidas y ya podemos hacer todo tipo de consultas.

3.1.6 Git

- ¿Qué es?



Git es un sistema de control de versiones distribuido gratuito y de código abierto diseñado para manejar todo, desde proyectos pequeños hasta muy grandes, con rapidez y eficiencia. Git es fácil de aprender y tiene una huella pequeña con un rendimiento increíblemente rápido.

Supera a Subversion, CVS, Perforce y ClearCase con características como ramificación local y múltiples flujos de trabajo.

- ¿Cómo lo he utilizado?

Me he creado un repositorio privado en Github al cual solo tiene acceso mi tutor y yo. Lo he utilizado básicamente para llevar un control de las tareas que tenía que hacer, las que estaba haciendo y cuales había terminado. También por cada tarea, creaba una rama local para yo poder trabajar

tranquilo en caso de que tuviera un error y no supiera cómo solucionarlo. Más adelante comentaré con más detalle el funcionamiento del repositorio en el apartado de Metodología.

- ¿Por qué lo he utilizado?

Como sistema de control de versiones he decidido utilizar Git básicamente porque es muy sencillo de utilizar y aprendí un montón en la asignatura de Metodologías Ágiles de Diseño Software. En este aspecto, mi tutor ha sido mucho más flexible, al igual que en los casos anteriores lo he querido utilizar para intentar aprender nuevas cosas, ya que hoy en día es muy raro no encontrarse con sistemas de control de versiones (VCS) sobretodo en proyectos de uso profesional. Sí que es verdad, que al estar yo solamente desarrollando el proyecto, no he podido profundizar mucho y aprender nuevas cosas ya que lo que necesitaba aplicar ya lo conocía anteriormente.

Es muy útil para comprobar si llevo arrastrando un error desde versiones anteriores, lo puedo identificar de forma rápida y sin perder funcionalidades que ya tenía previamente implementadas. Puedo llevar un control de los cambios que se van realizando y el momento que se hicieron.

Por ello, concluyo que si trabajas solo, usar Git es útil, pero si lo haces en equipo es fundamental.

3.1.7 Bullzip

- ¿Qué es? [5]



Es un programa que me va a permitir convertir mi base de datos que se encuentra en formato Access a MySQL. Presenta las siguientes características:

- Interfaz de asistente.
- Transfiere datos directamente de un servidor a otro.
- Crea un archivo de volcado.
- Permite seleccionar las tablas que quieres transferir.
- Permite seleccionar los campos que quieres transferir.
- Transfiere bases de datos protegidas con contraseña.
- Admite seguridad compartida y seguridad a nivel de usuario.
- Transferencia opcional de índices.
- Transferencia opcional de registros.
- Transferencia opcional de valores predeterminados en definiciones de campo.
- Identifica y transfiere los tipos del campo de número automático.
- Interfaz de línea de comando.
- Fácil instalación, desinstalación y actualización.

- ¿Cómo lo he utilizado?

La utilización de este programa es muy sencillo, lo primero que he hecho ha sido bajarme el programa e instalarlo en mi ordenador. En segundo lugar, he introducido la dirección origen donde se localiza el archivo de base de datos en formato Access. Finalmente, he indicado la dirección destino del nuevo archivo en formato SQL con todas las tablas seleccionadas las cuales quería transformar y el nombre de la base de datos.

- ¿Por qué lo he utilizado?

En este caso, he decidido utilizar este programa por todas las características comentadas anteriormente en la descripción del programa. Encontré algunos programas online que también realizaban el cambio, pero no lo hacían igual de eficiente y presentaba muchos errores, más adelante en el apartado de *Problemas y Soluciones* comentaré todos los problemas que tuve al realizar la conversión en los programas online y cómo este programa me ha solucionado la gran mayoría de problemas que tenía en el caso del online.

Capítulo 4

Metodología

En primer lugar, comentar que he utilizado metodologías ágiles para implementar el proyecto. A continuación, explicaré las razones que me llevaron a utilizarlas y cómo las he utilizado.

4.1 ¿Qué es la metodología ágil?

La metodología ágil surge como sustituta a los métodos clásicos de gestión. La flexibilidad, calidad y la necesidad de entregar proyectos y productos en cortos espacios de tiempo son una prioridad. La metodología ágil apuesta por un desarrollo en iteraciones cortas, cada iteración corta incorpora todas las fases del ciclo de vida:

- **Análisis de requisitos:** recogida de requisitos necesarios que debe cumplir el software.
- **Diseño:** estudio y descomposición de esos requisitos en partes a implementar.
- **Codificación:** implementación de cada una de las partes del diseño.
- **Integración y prueba:** integración del desarrollo en el producto y prueba de que se cumplen los requisitos.

Cada iteración tiene como resultado un producto al que se le han añadido nuevas características que pueden ser probadas por el cliente (incrementos), el

producto irá evolucionando y adoptando los cambios surgidos por el cliente hasta conseguir un producto final que lo satisfaga.








4.2 ¿Cómo las he utilizado?

Las metodologías ágiles que he utilizado han sido las siguientes.

4.2.1 *Scrum*

Se han realizado *sprints* semanales con mi tutor, cada semana obtenía un incremento del producto (conjunto de características adicionales que antes no existían y que se integran correctamente con las que ya existían) el cual mi tutor probaba todas las nuevas incorporaciones y elaboramos el sprint de la semana siguiente. Un ejemplo de cómo organizo el *backlog* del sprint en Github es el siguiente:

Figura 4.1: Ejemplo de listado de issues de Github

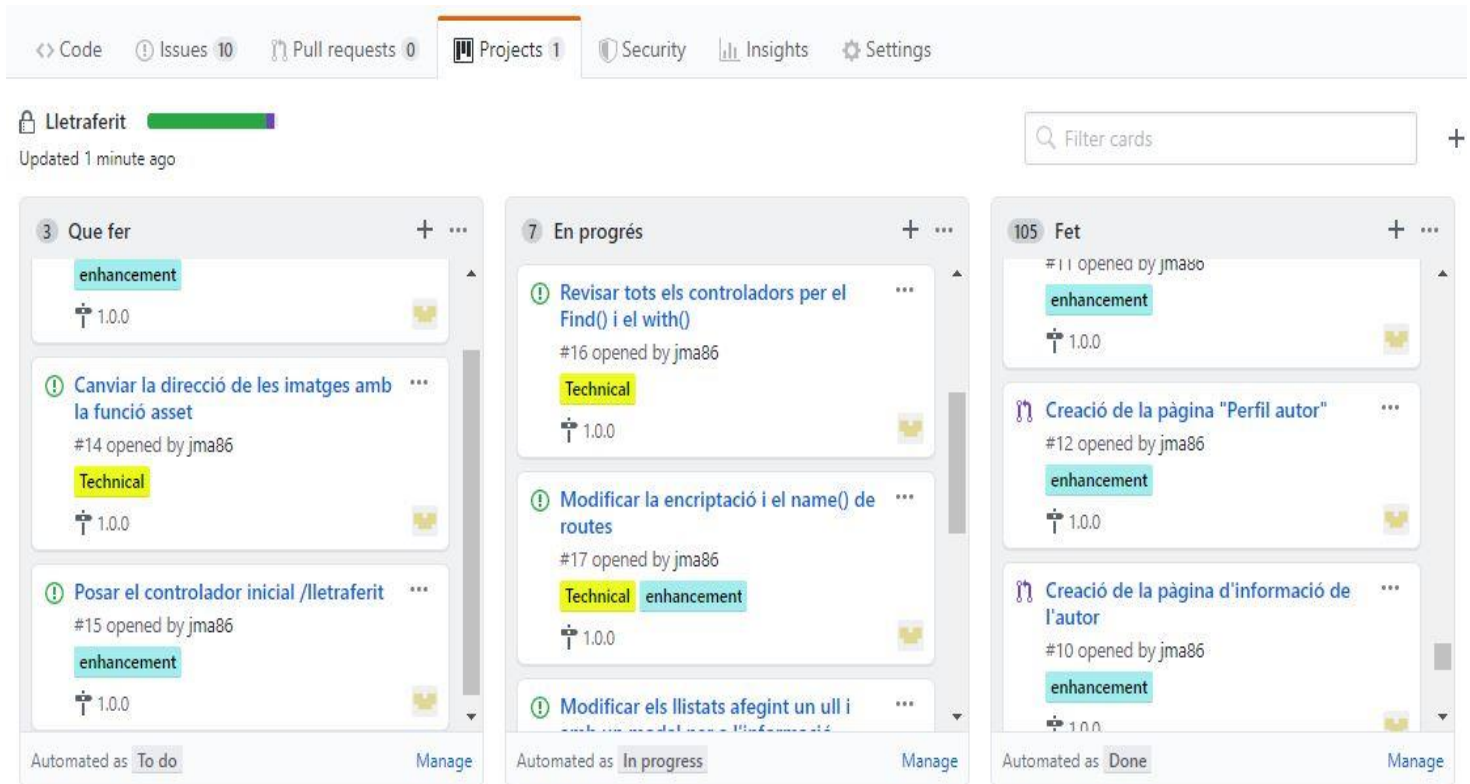
<input type="checkbox"/>	! Revisar tab i pàgina de resultats Technical	
	#22 opened on 11 Sep by jma86 1.0.0	
<input type="checkbox"/>	! Revisar la funció per obtenir la professió d'un autor bug	
	#21 opened on 11 Sep by jma86 1.0.0	
<input type="checkbox"/>	! Si la informació esta buida tindrem que ocultarla Technical	
	#20 opened on 11 Sep by jma86 1.0.0	
<input type="checkbox"/>	! Modificar els llistats amb un striped i afegir el withPivot Technical enhancement	
	#19 opened on 11 Sep by jma86 1.0.0	
<input type="checkbox"/>	! Modificar els llistats afegint un ull i amb un modal per a l'informació enhancement	
	#18 opened on 11 Sep by jma86 1.0.0	
<input type="checkbox"/>	! Modificar la encriptació i el name() de routes Technical enhancement	
	#17 opened on 11 Sep by jma86 1.0.0	
<input type="checkbox"/>	! Revisar tots els controladors per el Find() i el with() Technical	
	#16 opened on 11 Sep by jma86 1.0.0	
<input type="checkbox"/>	! Posar el controlador inicial /lletraferit enhancement	
	#15 opened on 11 Sep by jma86 1.0.0	
<input type="checkbox"/>	! Canviar la direcció de les imatges amb la funció asset Technical	
	#14 opened on 11 Sep by jma86 1.0.0	

Elaboración propia

4.2.2 Kanban

Durante el desarrollo del proyecto, me he creado un tablero *Kanban* en el Github para que pueda llevar un control de lo que me falta por hacer, lo que se está haciendo y lo que está terminado. Mi tablero está representado de la siguiente forma:

Figura 4.2: Tablero Kanban del proyecto creado en Github



Elaboración propia

El tablero está configurado de la siguiente forma:

- Cuando yo creo un *issue* automáticamente se coloca en la columna de *To do*.
- Cuando empiezo a trabajar sobre un *issue* debo moverlo a la columna de *In progress*.
- Cuando creo un *Pull request* y cierro el *issue* con la etiqueta Closes *#idIssue*, el *issue* se mueve automáticamente a la columna de *Done*.

Por último, las ramas que he utilizado han sido las siguientes:

- Master, de largo recorrido.
- issXXX (para cada *feature*).

4.3 ¿Por qué las he utilizado?

- *Scrum*

He utilizado *Scrum* básicamente para tener un control semanal de las tareas que voy haciendo e integrando bien, además de que mi tutor pueda cada semana ver cómo va evolucionando el proyecto. Si lo dejara todo para las últimas semanas, los cambios que tendría que realizar supondría un elevado coste en tiempo en comparación a si los hubiera realizado con anterioridad. Me ha resultado muy cómodo el poder realizar *sprints* semanales con tareas cortas y alguna con mayor dificultad. Fue una buena decisión ya que en mi opinión estancarse 2 o 3 semanas en una misma tarea hubiera sido un trabajo bastante pesado. Con la ayuda de Github, he podido organizar las tareas de una forma sencilla en la que no he tenido ningún problema a la hora de implementarlas.

- *Kanban*

Decidí utilizar un tablero *Kanban* en Github por qué si no lo hubiera utilizado, si pasan un par de días y no me acuerdo de lo que estaba hecho y lo que no, podría dejarme cosas sin hacer o hacer cosas que no son tan importantes. Por ello, me cree un tablero *Kanban* en Github con tres columnas comentadas anteriormente, donde podía organizarme todas las tareas pendientes que me faltaban, cuáles eran las más importantes y cuales estaban terminadas. Al igual que en el caso anterior, no he tenido ningún problema a la hora de saber el estado de cada tarea.

Capítulo 5

Análisis

Como he comentado al inicio, antes de empezar con la implementación de la web he realizado un análisis del proyecto. En el análisis he identificado las siguientes tareas:

- Convertir la base de datos de Access a MySQL.
- Cambios en la base de datos.
- Diseñar los diagramas de ER.
- Implementar los *mockups*.
- Crear el repositorio de Github y el proyecto de Laravel.

5.1 Convertir la base de datos de Access a MySQL

En primer lugar, necesitaba un programa que pudiera realizar el cambio, con la posibilidad de no perder los datos para que pueda observar todas las tablas con las que voy a trabajar y realizar los cambios oportunos. Tras probar varios programas, decidí quedarme con el Bullzip ya que era el que mejor conversiones hacía de este tipo.

5.2 Cambios en la base de datos

Los cambios se realizarán por dos razones: por peticiones de los usuarios y por adaptaciones necesarias al actualizar el sistema a la nueva tecnología.

Tras adaptar el sistema a la nueva tecnología, he detectado que se necesitan realizar una serie de cambios importantes en la base de datos. Los cambios realizados en las tablas existentes han sido los siguientes:

Tabla 5.1: Nuevos campos en las tablas

Tabla	Campo	Tipo	Propósito
1mautorcurt / eautors	sexe	char(2)	Se tiene que poder identificar o buscar los autores que son hombres y los que son mujeres.
	foto	varchar(255)	Posibilidad de subir una foto de perfil.
1meditors	direccio	varchar(255)	Posibilidad de observar la editorial de forma electrónica.
1mid idioma	imatge	varchar(255)	El idioma estará identificado también por la imagen de su bandera.
eautors	blog	varchar(255)	Mostrar la dirección del blog del autor si lo tuviera.
ecritiques / epublicacio	link	varchar(255)	Mostrar la dirección de la crítica/publicación de forma

			electrónica.
	electronic	char(2)	Indicará con una ‘S’ si la crítica/publicación se puede consultar de forma electrónica o ‘N’ en caso contrario.
eobres	image	varchar(255)	Posibilidad de subir una foto de perfil de la obra.
	musica	varchar(255)	Archivo en formato .mp3 con la música de la obra.
	so	varchar(255)	Indicará con una ‘S’ si la obra tiene música o ‘N’ en caso contrario.

Elaboración propia

Tabla 5.2: Campos eliminados

Tabla	Campo	Tipo	Propósito
eobres	bua	varchar(5)	No aporta información relevante.

Elaboración propia

Tabla 5.3: Nuevas tablas con sus campos y relaciones

Tabla	Campos	Tipo	Observaciones
cms	id	Primary Key (int(11))	Para almacenar el contenido de la páginas estáticas.
	txt	text	
	url	varchar(255)	
model_has_permissions	permission_id	Foreign Key (int(10) -> permissions (id / int(10)))	Indica los accesos del modelo.
	model_id	Primary Key (bigint(20))	
	model_type	Primary Key (varchar(255))	
model_has_roles	role_id	Foreign Key (int(10) -> roles (id / int(10)))	Indica los roles del modelo.
	model_type	Primary Key (varchar(255))	
	model_id	Primary Key (bigint(20))	
password_resets	email	varchar(255)	Guardar

	token	varchar(255)	contraseñas restablecidas.
	created_at	timestamp	
permissions	id	Primary Key (int(10))	Indica los posibles accesos.
	name	varchar(255)	
	guard_name	varchar(255)	
	created_at	timestamp	
	updated_at	timestamp	
role_has_permissions	permission_id	Foreign Key (int(10) -> permissions (id / int(10)))	Indica los accesos de cada role.
	role_id	Foreign Key (int(10) -> roles (id / int(10)))	
roles	id	int(10)	Indica los posibles roles.
	name	varchar(255)	
	guard_name	varchar(255)	
	created_at	timestamp	
	updated_at	timestamp	
subscribers	id	int(11)	Guarda los

	nom	varchar(255)	usuarios suscritos a la web.
	correu	varchar(255)	
users	id	int(10)	Usuarios para el <i>back office</i> .
	name	varchar(255)	
	email	varchar(255)	
	email_verified_at	timestamp	
	password	varchar(255)	
	remember_token	varchar(100)	
	created_at	timestamp	
	updated_at	timestamp	

Elaboración propia

Tabla 5.4: Tablas eliminadas

Tabla	Observaciones
config	No es necesaria.

Elaboración propia

Nota: Tablas que existían en el anterior proyecto y que en la modernización no son necesarias.

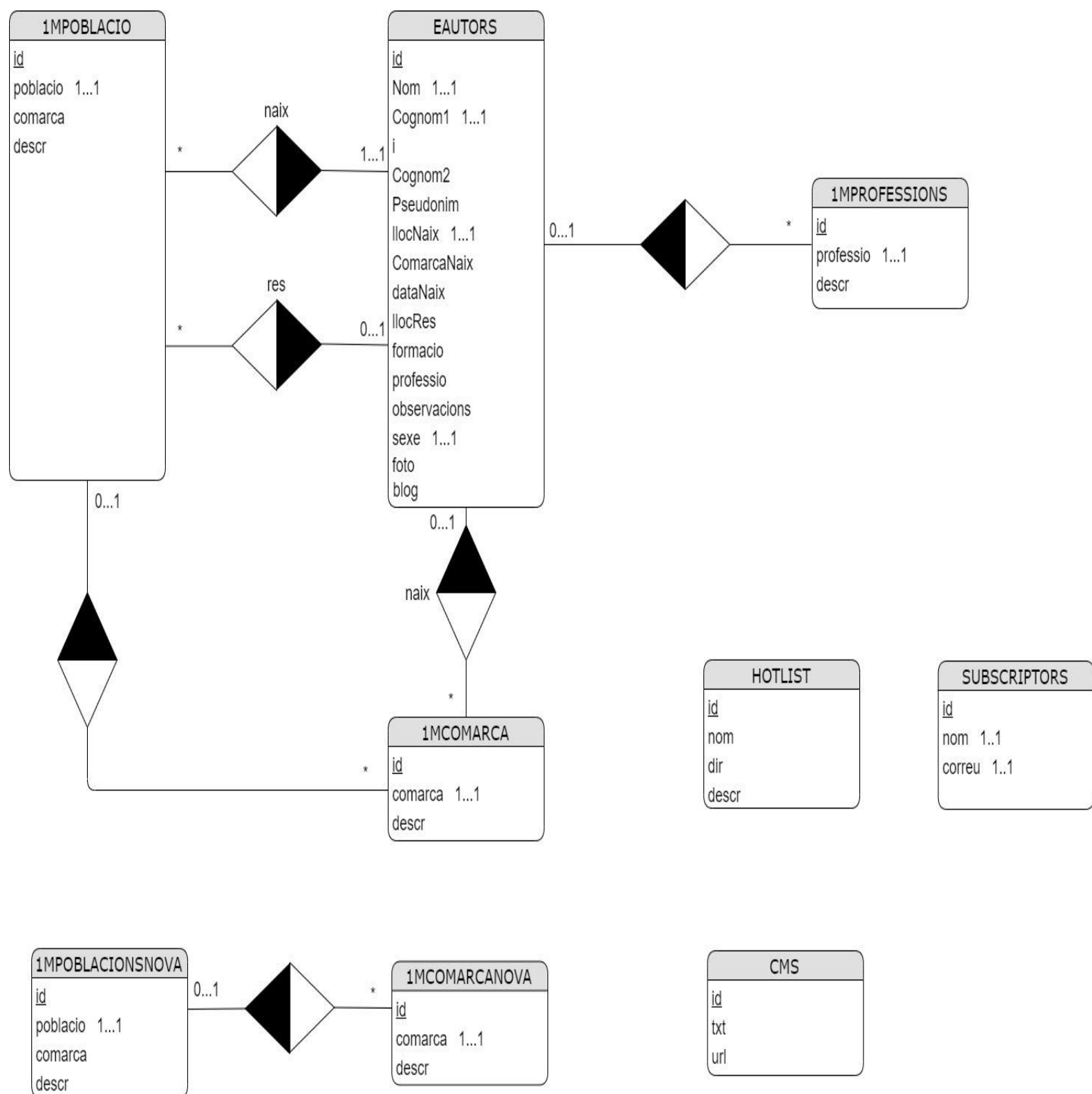
Más adelante, comentaré todos y cada uno de los problemas que tuve tanto en la conversión como en los diferentes valores de la base de datos.

5.3 Diseñar los diagramas de ER

Un diagrama ER (Entidad-Relación) es un tipo de diagrama que ilustra cómo las entidades (objetos principales del sistema) se relacionan entre sí dentro del proyecto. Una vez he convertido la base de datos y he realizado las modificaciones pertinentes, tocaba obtener los diagramas de acuerdo a como estaba representada la base de datos del antiguo proyecto (proceso que se conoce como ingeniería inversa). Este apartado es muy importante ya que si no se plantean bien, estaré continuamente haciendo cambios con la consecuente pérdida de tiempo que esto provoca.

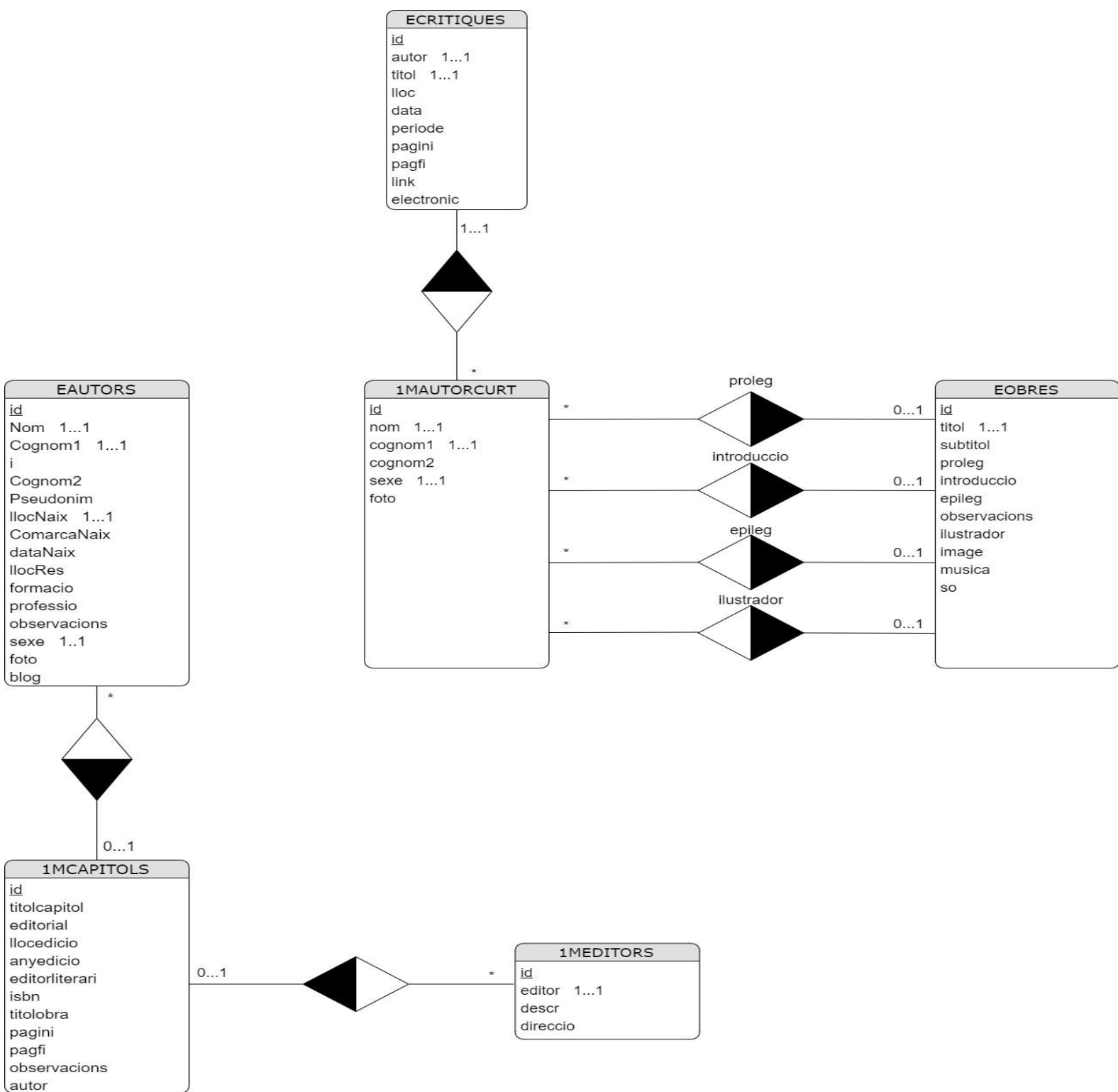
Para los diagramas de ER, he decidido separarlos en cuatro grandes grupos, ya que en caso contrario me quedaba un diagrama muy grande. Las particiones realizadas han sido las siguientes:

Figura 5.1: Diagrama de Eautors -> donde la tabla principal es la los autores



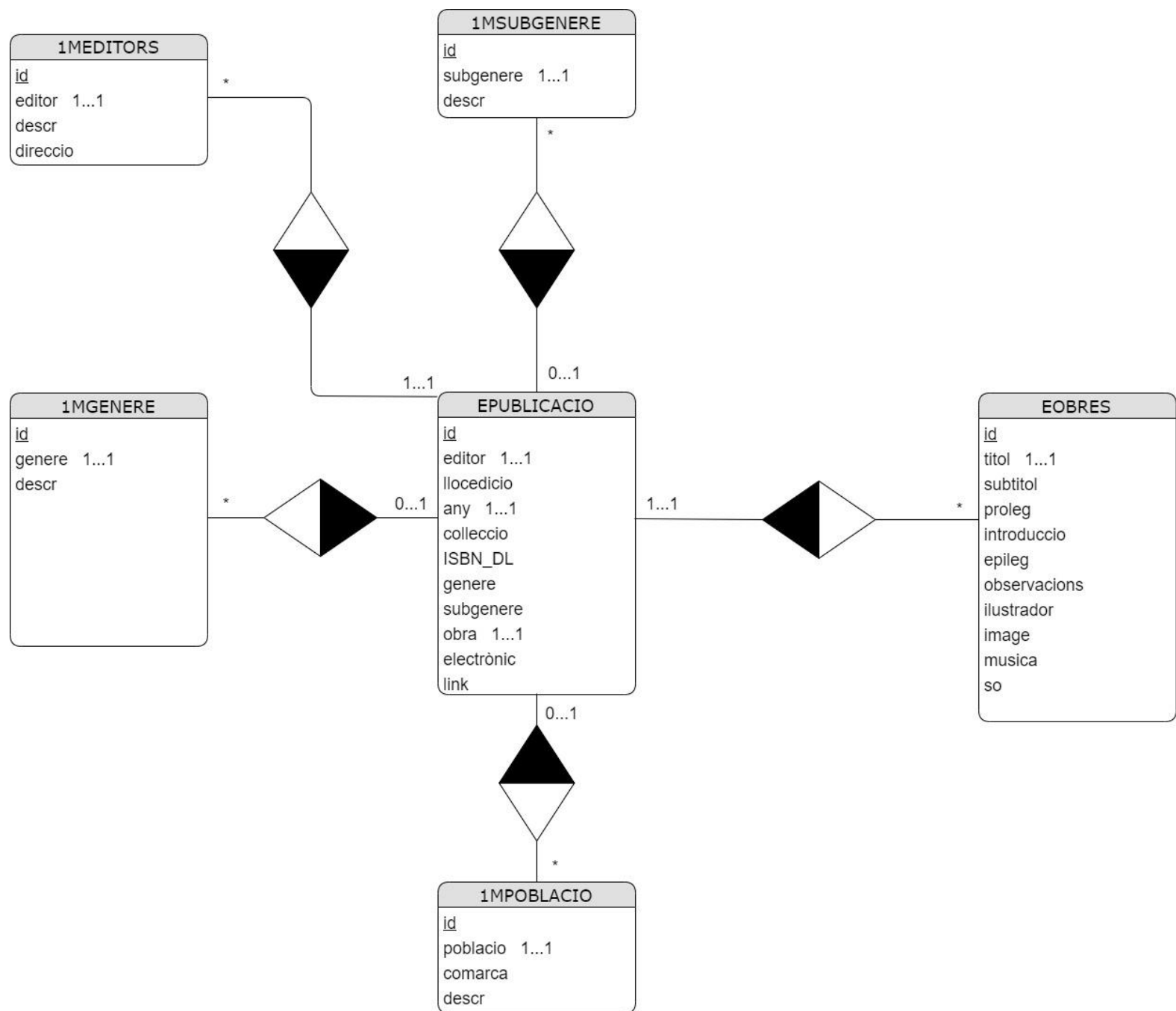
Elaboración propia

Figura 5.2: Diagrama de Eobres -> donde la tabla principal es la de obras



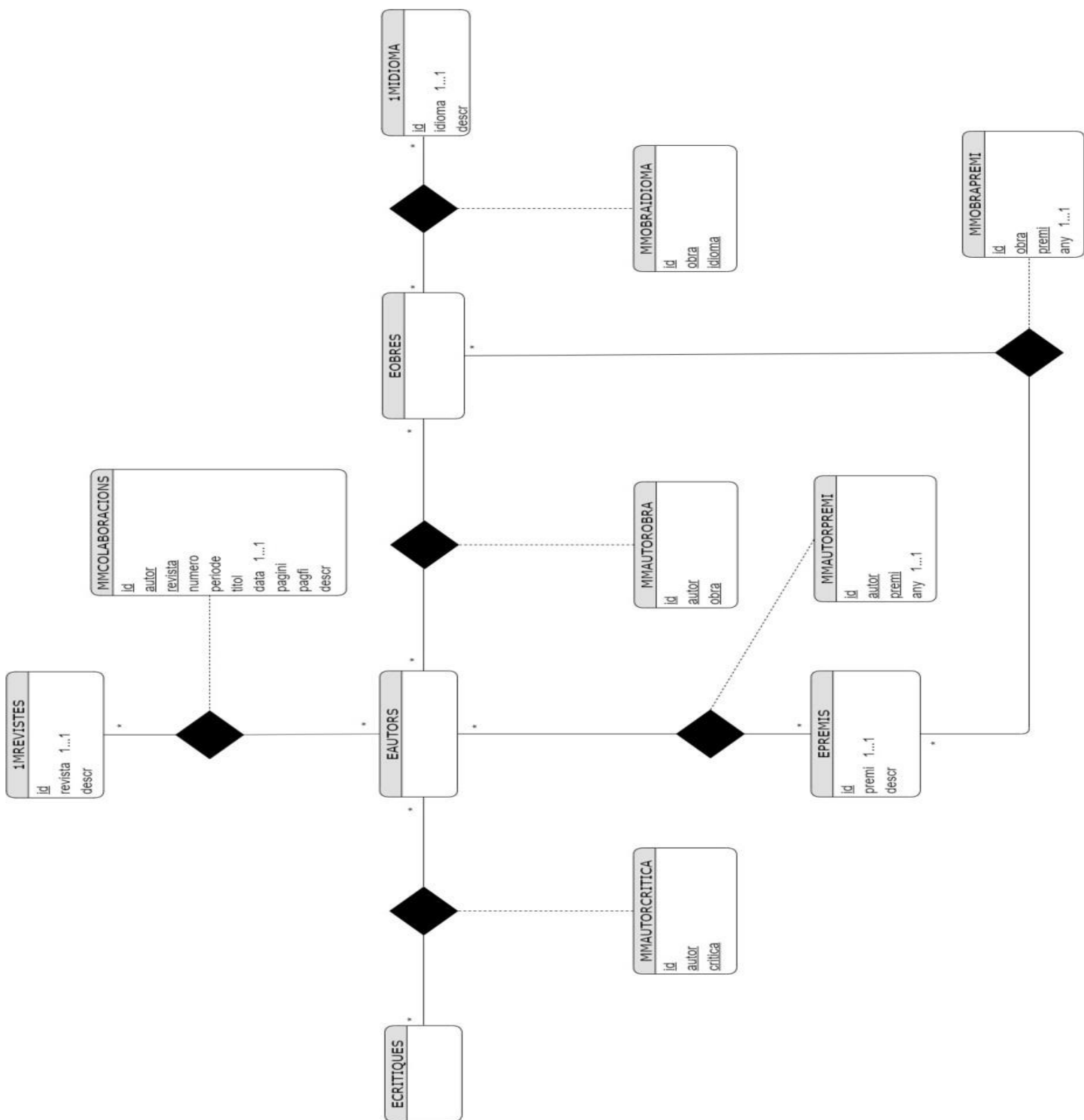
Elaboración propia

Figura 5.3: Diagrama de Epublicacio -> donde la tabla principal es la de publicaciones



Elaboración propia

Figura 5.4: Diagrama de MM -> donde aparecen todas las relaciones de M:M



Elaboración propia

Nota 1: Se han respetado los nombres de tablas y atributos conforme estaban en el proyecto antiguo.

Nota 2: Las entidades del último diagrama que aparecen vacías tienen los mismos campos que en los diagramas anteriores, he decidido eliminarlos en el último para optimizar espacio y visibilidad.

Nota 3: Para los diagramas anteriores, no he podido respetar los márgenes interior y exterior para su correcta visualización.

5.4 Implementar los *mockups*

Los *mockups* son fotomontajes que permiten a los diseñadores gráficos y web mostrar al cliente cómo quedarán sus diseños. Se desarrollarán los *mockups* con una propuesta mía y luego se mostrarán a los usuarios para que los validen.

Tras la primera reunión que tuve con mi tutor, me planteó un escenario en el que el usuario tuviera un menú horizontal con cuatro opciones:

- Presentació
- Descripció
- Investigadors
- Agraïments

Y luego que tuviera un buscador a modo de *Tabs*, en los que en cada uno debía de aparecer un botón para realizar la búsqueda. Los *tabs* acordados han sido los siguientes:

- Autors
- Obres
- Premis
- Altres

Por último, en cada búsqueda que realizaba, tenía que aparecer inicialmente un listado del resultado buscado. En cada listado, podía tener la posibilidad de consultar la información de un único elemento. [6] Con toda esta información, he diseñado los *mockups* de las páginas principales de la siguiente forma:

Figura 5.5: Página principal con el primer tab

Mockup of a web page titled "Figura 5.5: Página principal con el primer tab". The page is displayed in a Mozilla browser window with the URL "http://moqups.com".

The layout includes a header with "Xarxes socials", a placeholder for a logo, and "Contacta'ns: @correu".

Below the header is a navigation bar with tabs: "Presentació", "Descripció", "Investigació", and "Agraïments".

The main content area features a search section with tabs "AUTORS", "OBRES", "PREMIS", and "ALTRES". The "AUTORS" tab is active, showing a search form with fields for "Pseudonim", "nom + cognoms", and gender selection (M, F, A), followed by a "Buscar" button.

The footer contains a placeholder for a logo, "Contactar", "Tags", "Newsletter" (with a description "Lorem ipsum dolor sit amet, consectetur" and an "Email Address" input field), and "Xarxes socials".

Elaboración propia

Figura 5.6: Pàgina principal con el segundo tab

Mozilla

← → ↻

http://moqups.com

Xarxes socials

Contacta'ns: [@correu](#)

Presentació

Descripció

Investigació

Agraïments

AUTORS

OBRES

PREMIS

ALTRES

Títol

Buscar

Contactor

Tags

Newletter

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate

Email Address

Xarxes socials

Elaboración propia


Figura 5.7: Pàgina principal con el tercer tab

Mozilla

← → ↻

http://moqups.com

Xarxes socials



Contacta'ns: [@correu](#)

Presentació

Descripció

Investigació

Agraïments

AUTORS


OBRES

PREMIS

ALTRES

Nom del premi

Buscar



Lorem ipsum dolor sit
amet, consectetur
adipiscing elit. Nulla
quam velit. vulbutate

Xarxes socials

Contactar

Tags

Newletter

Lorem ipsum dolor sit
amet, consectetur

Email Address

Elaboración propia


Figura 5.8: Pàgina principal con el cuarto tab

Mozilla

← → ↻

http://moqups.com

Xarxes socials



Contacta'ns: [@correu](#)

Presentació

Descripció

Investigació

Agraïments


AUTORS

OBRES

PREMIS

ALTRES

Buscar



Lorem ipsum dolor sit
amet, consectetur
adipiscing elit. Nulla
quam velit. vulbutate

Xarxes socials

Contactar

Tags

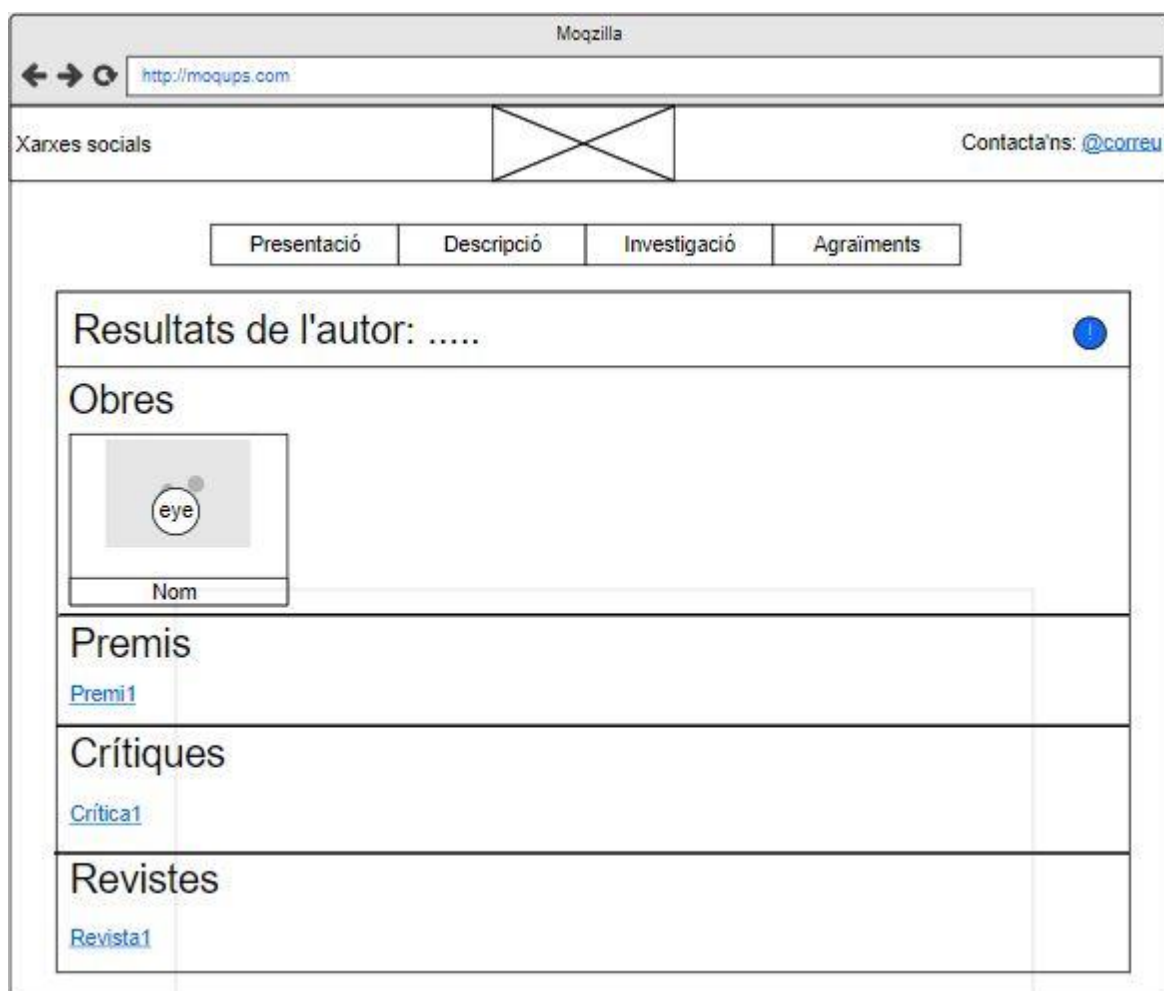
Newletter

Lorem ipsum dolor sit
amet, consectetur

Email Address

Elaboración propia

Figura 5.9: Pàgina de informació del autor



Elaboración propia

Figura 5.10: Pàgina de perfil del autor

Moqzilla

← → ↻

http://moqups.com

Xarxes socials

Contacta'ns: @correu

Presentació

Descripció

Investigació

Agraïments

Nom

Cognom1

Cognom2

Pseudonim

⋮

Obres

Nom

Contactor

Tags

Newletter

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit vulputate

Email Address


Xarxes socials

Elaboración propia


Figura 5.11: Pàgina de informació de una obra

Moqzilla

← → ↻ <http://moqups.com>

Xarxes socials  Contacta'ns: [@correu](#)

Presentació	Descripció	Investigació	Agraïments
-------------	------------	--------------	------------



- Títol de l'obra
- Subtítol
- Proleg
- Introducció
- Epíleg
- Il·lustrador

Observacions

Premis aconseguits

[Premi1](#)

Idiomes

[Idioma1](#)

Elaboración propia

Figura 5.12: Pàgina de informació de un idioma

Mozilla

← → ↻

http://moqups.com

Xarxes socials

Contacta'ns: @correu

Presentació

Descripció

Investigació

Agraïments

Idioma:

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Nulla quam velit, vulputate eu
pharetra nec. mattis ac neque.

Obres amb el idioma

Nom

Contactar

Tags

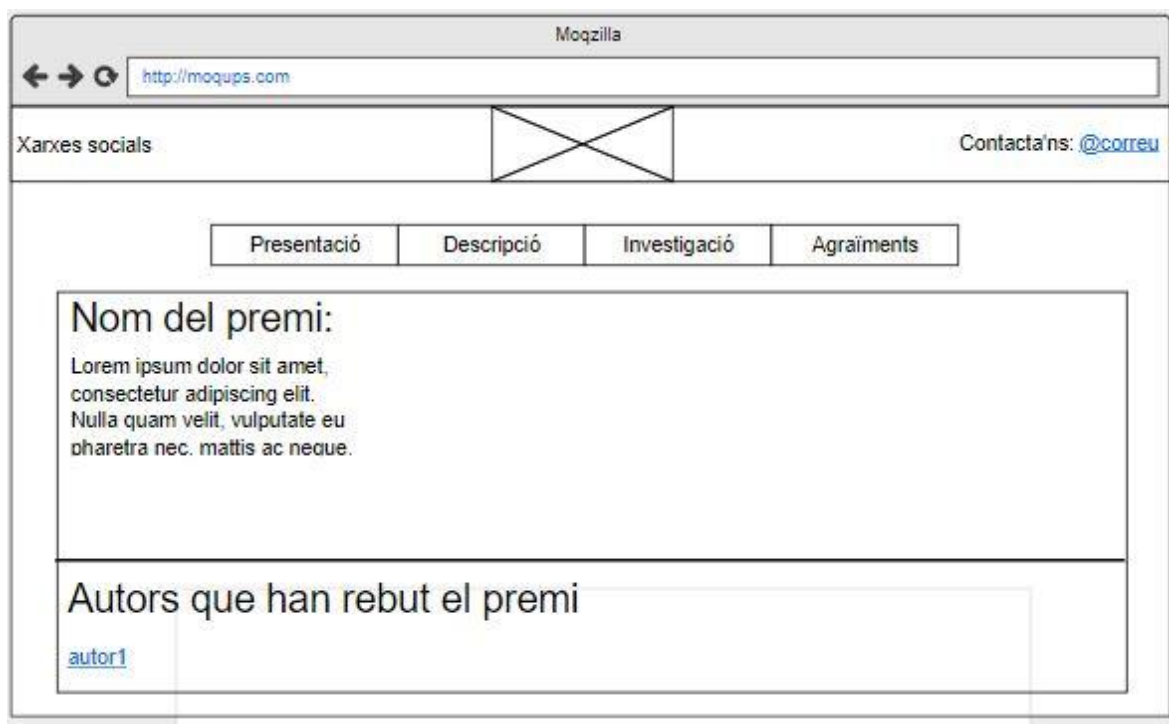
Newsletter

Lorem ipsum dolor sit
amet, consectetur

Xarxes socials

Elaboración propia

Figura 5.13: Página de información de un premio



Elaboración propia

Comentar que estas páginas como es habitual pueden ser modificadas a lo largo del proyecto, es una primera idea en la cual me permite poder empezar a implementar el proyecto.

5.5 Crear el repositorio de Github y el proyecto de Laravel

Por último, se creará un repositorio en Github privado, el cual solo tendrá acceso mi tutor. Como primer *commit*, se subirá la creación de un proyecto de Laravel de la versión 5.5 con un *.gitignore*.

Capítulo 6

Implementación

Una vez realizadas todas las tareas anteriores, al igual que en la fase anterior he identificado las tareas que voy a llevar a cabo durante la fase de implementación:

- Implementar el *front office*
- Subir la web a producción
- Implementar el *back office*

6.1 Implementar el *front office*

Una vez completado todos los puntos del análisis, empecé a implementar el sitio web. Lo primero que hice, fue intentar buscar una plantilla HTML que se pareciera a los *mockups* diseñados anteriormente. Tuve mucha suerte y encontré una plantilla llamada [8] “*Callie*” la cual se parecía un montón al diseño de mis *mockups*. Tras incluir la plantilla y modificarla con lo que necesitaba, el resto del trabajo lo abordé de la siguiente forma:

- A partir de los *mockups* anteriores preparaba una plantilla de vista (HTML y elementos necesarios).
- A continuación, para cada *mockup* individual me centraba en su contenido específico. Por ejemplo, si estaba implementando una vista

de un *tab* principal, si ese *tab* tenía una lista desplegable o un cuadro para poder introducir texto.

- Cada vez que necesitaba datos (consultas) creaba un controlador y le añadía las acciones necesarias. Este controlador, una vez obtenía los datos que necesitaba el usuario, devolvía la vista con los datos.
- Para cada acción, creaba su ruta pertinente, la cual siempre iba asociada a un controlador. A medida que el usuario iba visitando las diferentes rutas, se iban llamando a los diferentes controladores.
- A veces, veía que podía reutilizar código y acciones, por eso encapsulaba el código para hacerlo reutilizable. Por ejemplo, para consultar una única obra desde el *Tab*, reutilizo una función del controlador porque en mi caso, al implementar el primer *Tab* (Index autors/es) dentro de la información del autor, el usuario tiene la posibilidad de acceder a la información de una obra, por tanto cuando empecé a implementar el tercer *Tab* (Obres) me di cuenta que la vista era la misma. Al implementar la función, lo único nuevo que hacía era comprobar si el usuario había seleccionado alguna obra, si todo era correcto llamaba a la función que obtenía toda la información de una obra.

Uno de los puntos importantes en la implementación del proyecto ha sido la utilización de los controladores, ya que me permite agrupar toda la lógica de una funcionalidad en una sola clase. Los controladores se almacenan en el directorio *app/Http/Controllers*. En mi caso, los controladores que he utilizado han sido los siguientes:

Tabla 6.1: Información del controller AltresAutorsController

AltresAutorsController	
Descripción: controla las opciones del tab Altres Consultes / Altres autors.	
Función	Acción
consultarAltresAutors(\$nom)	Cuando seleccionamos una de las cuatro opciones, carga el nombre y apellidos de los autores/as en la <i>selec</i> dependiendo del botón pulsado.
consultarAltreAutor(Request \$request)	Obtiene toda la información del autor/a seleccionado cuando pulsamos el botón de buscar. Si no ha seleccionado ningún autor/a muestra un error.

Elaboración propia

Tabla 6.2: Información del controller CapsaleraController

CapsaleraController	
Descripción: controla las cuatro opciones del apartado “Informació” del footer.	
Función	Acción
presentacio()	Muestra la página estática “presentación”.
descripcio()	Muestra la página estática “descripción”.

investigacio()	Muestra la página estática “investigación”.
agreements()	Muestra la página estática “agradecimientos”.

Elaboración propia

Tabla 6.3: Información del controller ConsultaAltresController

ConsultaAltresController	
Descripción: controla las opciones del tab Altres Consultes / Consulta obres de teatre infantil, Consulta obres de teatre juvenil, Consulta obres de literatura infantil, Consulta obres de literatura juvenil.	
Función	Acción
consultaTeatreInfantil()	Devuelve todas las obras de teatro infantil.
consultaTeatreJuvenil()	Devuelve todas las obras de teatro juvenil.
consultaLiteraturaInfantil()	Devuelve todas las obras de literatura infantil.
consultaLiteraturaJuvenil()	Devuelve todas las obras de literatura juvenil.

Elaboración propia

Tabla 6.4: Información del controller ConsultaObraController

ConsultaObraController	
Descripción: controla las opciones del tab Altres Consultes / Consulta obres amb música, Consulta obres electròniques, Consulta obres en col·laboració. Además controla la selección de una obra a la hora de buscar.	
Función	Acción
consultarObra(\$idObra)	Devuelve toda la información de una obra.
consultarObraTab()	Controla en el <i>tab</i> Obres que se seleccione alguna obra al pulsar el botón de búsqueda, si todo es correcto llama a la función anterior.
consultaObresColaboracio()	Devuelve todas las obras escritas por más de un autor.
consultaObresMusica()	Devuelve todas las obras con música.
consultarObresElectroniques()	Devuelve todas las obras que se pueden consultar de forma electrónica.

Elaboración propia

Tabla 6.5: Información del controller ContactarController

ContactarController	
Descripción: controla el hiperenlace superior izquierdo: “Actualització de dades” donde los usuarios pueden informar al administrador de posibles errores en la información.	
Función	Acción
actualitzacio()	Envía al usuario a la vista del formulario de incidencias.
comprobarActualitzacio(Reques \$request)	Comprueba que todos los datos escritos por el usuario en el formulario son correctos y devuelve un mensaje.

Elaboración propia

Tabla 6.6: Información del controller EditorialController

EditorialController	
Descripción: controla la opción del tab Altres Consultes / Editorials.	
Función	Acción
consultarEditorials(Request \$request)	Devuelve toda la información de la editorial cuando el usuario selecciona una editorial y pulsa el botón de buscar. Controla si el usuario selecciona una editorial, en caso contrario muestra un mensaje de error.

Elaboración propia

Tabla 6.7: Información del controller IlustradorController

IlustradorController	
Descripción: controla la opción del tab Altres Consultes / Il·lustradors.	
Función	Acción
consultaIlustrador(Request \$request)	Devuelve toda la información del ilustrador cuando el usuario selecciona un ilustrador y pulsa el botón de buscar. Controla si el usuario selecciona un ilustrador, en caso contrario muestra un mensaje de error.

Elaboración propia

Tabla 6.8: Información del controller IndexController

IndexController	
Descripción: controla todos los listados que hay en las opciones del <i>Tab</i> y Cercadors.	
Función	Acción
resultatsAutors()	Devuelve todos los autores/as que contengan la información introducida por el usuario.
resultatsObres()	Devuelve todas las obras que contengan la información introducida por el usuario.
resultatsPremis()	Devuelve todos los premios que contengan la información introducida por el usuario.
genere(\$idGenere)	Devuelve todas las obras del género seleccionado.
genereIsubgenere()	Devuelve todas las obras del género y subgénero elegidos por el usuario. Obligatoriamente, el usuario debe de seleccionar un género, el subgénero si no lo selecciona no pasa nada.
idioma(\$idIdioma)	Devuelve todas las obras del idioma seleccionado.
resultatsPublicacions()	Devuelve todas las publicaciones que contengan la información introducida por el usuario.

Elaboración propia

Tabla 6.9: Información del controller PremiController

PremiController	
Descripción: controla la opción del tab Altres Consultes / Premis.	
Función	Acción
info(\$idPremi)	Devuelve toda la información del premio que recibe por parámetro.
consultarPremiTab(Request \$request)	Busca el premio seleccionado por el usuario y llama a la anterior función. En caso de que el usuario no seleccione algún premio, muestra un mensaje de error.

Elaboración propia

Tabla 6.10: Información del controller ResultatAutorController

ResultatAutorController	
Descripción: controla la información de un autor.	
Función	Acción
resultats(Request \$request)	Devuelve toda la información del autor/a seleccionado por el usuario y controla si se ha seleccionado un autor.

Elaboración propia

Tabla 6.11: Información del controller *TabsController*

TabsController	
Descripción: controla el funcionamiento del <i>Tab</i> , además del abecedario de cada opción del <i>tab</i> .	
Función	Acción
tabs()	Cuando se inicia la web, carga todo el contenido necesario para los despleables de las opciones del <i>Tab</i> .
abecedariAutors(\$lletra)	Cuando el usuario selecciona una letra, filtra por el primer apellido todos los autores/as que empiezan por esa letra.
abecedariObres(\$lletra)	Mismo proceso que el caso anterior pero por el título de la obra.
abecedariPremis(\$lletra)	Mismo proceso que el caso anterior pero por el nombre del premio.
abecedariIl·lustradors(\$lletra)	Mismo proceso que el autor.
abecedariEditorials(\$lletra)	Mismo proceso que el caso anterior pero por el nombre de la editorial.
generesDestacats()	Función que selecciona los cinco géneros con más obras y por Ajax se introducen en el apartado “Generes destacats” del <i>footer</i> .
fullInformatiu(Request \$request)	Función que comprueba los datos introducidos del apartado “Subscriu-te al full informatiu” y

	realiza un <i>insert</i> a la tabla <i>subscriptors</i> si los datos son correctos.
--	---

Elaboración propia

Lógicamente para cada controlador/acción definidas anteriormente, he tenido que crear una ruta que le dé acceso desde HTTP. Las rutas, han quedado de la siguiente forma:

Tabla 6.12: Rutas principales del sitio web

Funcionalidad	Ruta	Método	Destino (controlador/accion)
Página principal	/ (raiz)	Get	Tabs / tab
Abecedario del tab Índex autors/es	/abecedariAutors/{lletra}	Get	Tabs / abecedariAutors
Obtener géneros destacados	/generesDestacats	Get	Tabs / generesDestacats
Obtener información de un autor	/resultats	Post	ResultatAutor / resultats

Consultar la información de una obra	/consultaObra/{id}	Get	ConsultarObra / consultarObra
Controlar errores al seleccionar una obra	/seleccioObra	Post	ConsultarObra / consultarObraTab
Devuelve un listado de autores	/indexAutors	Get	Index / resultatsAutors
Devuelve un listado de obras dependiendo del género seleccionado	/obresGenere/{id}	Get	Index / genere
Comprobar datos del formulario “Actualització de dades”	/comprobarActualitzacio	Post	Contactar / comprobarActualitzacio
Obtener información de un ilustrador y controlar errores	/seleccioIlustrador	Post	Ilustrador / consultarIlustrador

Devuelve un listado de las obras de teatro infantil	/consultaTeatreInfantil	Get	ConsultaAltres / consultaTeatreInfantil
Comprobar datos y subscribirse en caso afirmativo	/fullInformatiu	Post	Tabs / fullInformatiu
Devuelve toda la información de otros autores/as	/consultarAltreAutor	Post	AltresAutors / consultarAltreAutor

Elaboración propia

Una de las partes más importantes en el desarrollo del proyecto, ha sido la implementación del abecedario de forma dinámica para cada opción del *Tab*, el código utilizado ha sido el siguiente:

Figura 6.1: Fichero tabs donde se carga el contenido del abecedario dependiendo del tab seleccionado

```
<form method="POST" action="{ { helper::myUrl('/resultats') } }" role="form">
@include('abecedari.abecedari',['funcio' => 'autor'])
  {{ csrf_field() }}
  <select id="contingutAutor" name="autor" class="form-control">
    <option value="0">Selecciona una lletra del abecedari</option>
  </select>
  <div>
    <button type="submit" id="botoBusqueda" class="btn btn-success">
      <i class="fa fa-search"></i>
      Buscar
    </button>
  </div>
</form>
```

Elaboración propia

Inicialmente, se carga el contenido del abecedario, almacenado en otro fichero, en el fichero principal de *tabs* comprobando de qué parte del *tab* viene el usuario. Al cargar el contenido, se le pasa una variable *\$funcio* indicándole el *tab* del usuario, en este caso se trata del *tab* Autores/es. Una vez se comprueba de dónde viene el usuario, habrá una etiqueta *select* con un identificador *id* el cual tendrá una única opción por el momento.

Figura 6.2: Fichero abecedari con bucles for para construir los botones del abecedario

```
15 <div class="abecedari">
16   <ul>
17     <div class="col-md-12">
18       @for ($i = 65; $i <= 77; $i++)
19         <?php $lletra = chr($i); ?>
20         <li><button class="btn btn-outline-info" type="button" onclick="lletra('{{ $lletra }}', '{{ $funcio }}')">
21           {{ $lletra }}</button></li>
22       @endfor
23     </div>
24     <div class="col-md-12">
25       @for ($i = 78; $i <= 90; $i++)
26         <?php $lletra = chr($i); ?>
27         <li><button class="btn btn-outline-info" type="button" onclick="lletra('{{ $lletra }}', '{{ $funcio }}')">
28           {{ $lletra }}</button></li>
29       @endfor
30     </div>
31   </ul>
32 </div>
```

Elaboración propia

A continuación, utilizando el código ASCII se construyen los botones de la A - M y luego de la N - Z, lo he implementado de esta forma básicamente porque visualmente para el usuario era mucho mejor separarlo en dos líneas que no en una sola, ya que en caso contrario el usuario tenía que realizar un *scroll* horizontal. Dentro de cada *for*, vuelvo a utilizar la etiqueta de PHP para guardar en una variable la letra para luego poder pasarla al método de Javascript cuando el usuario pulse cualquier botón del abecedario.

Figura 6.3: Resultado del código de la Figura 6.1 y Figura 6.2

The image shows a web application interface with a dark header bar containing five navigation links: 'ÍNDIX AUTORS/ES' (highlighted in red), 'CERCADOR AUTORS/ES', 'OBRES', 'CERCADOR OBRES', and 'ALTRES CONSULTES'. Below the header, there is a grid of 26 buttons representing the alphabet (A-Z). Underneath the grid is a dropdown menu with the placeholder text 'Selecciona una lletra del abecedari'. At the bottom of the interface is a large green button with a magnifying glass icon and the text 'Buscar'.

Elaboración propia

Cuando el usuario selecciona cualquier botón del abecedario, se ejecuta el siguiente código:

Figura 6.4: Función Javascript que comprueba el origen del usuario, localiza el componente select y la función a invocar

```
40 function lletra(lletra,funcio) {
41
42     if(funcio == "autor") {
43         var capa = document.getElementById("contingutAutor");
44         var ulrFuncio = 'abecedariAutors/' + lletra;
45     }
46     if(funcio == "obra") {
47         var capa = document.getElementById("contingutObra");
48         var ulrFuncio = 'abecedariObres/' + lletra;
49     }
50     if(funcio == "premi") {
51         var capa = document.getElementById("contingutPremi");
52         var ulrFuncio = 'abecedariPremis/' + lletra;
53     }
54     if(funcio == "ilustrador") {
55         var capa = document.getElementById("contingutIlustrador");
56         var ulrFuncio = 'abecedariIlustradors/' + lletra;
57     }
58     if(funcio == "editorial") {
59         var capa = document.getElementById("contingutEditorial");
60         var ulrFuncio = 'abecedariEditorials/' + lletra;
61     }
}
```

Elaboración propia

Al igual que en el caso anterior, se comprueba que función se tiene que ejecutar dependiendo de la opción del *tab* en el que el usuario esté realizando la búsqueda, porque cada *select* está identificada con un id propio, entonces si no localizo el componente id, al realizar una búsqueda se introducirán los resultados en todos los *selects* y no se devolverá la información correcta.

Figura 6.5: Función AJAX que invoca la función localizada en la Figura 6.4 e inyecta el resultado en el componente select localizado en la Figura 6.4

```
63     $.ajax({
64         url: ulrFuncio,
65         type: 'get',
66         encode : true,
67         success: function(resultat) {
68             capa.innerHTML = resultat;
69         }
70     })
71 }
```

Elaboración propia

Una vez tenemos el *select* localizado y la función, realizo una llamada *AJAX* muy sencilla de tipo *GET* al *controller TabsController* cuya función se define en la variable *var ulrFuncio*.

Figura 6.6: Función que devuelve todos los autores cuya letra del primer apellido empiece por la letra pasado por parámetro

```
35 public function abecedariAutors($lletra) {  
36  
37     $autors = Eautor::where('Cognom1','like',$lletra . '%')->orderBy('Cognom1')->get();  
38  
39     $resultat = "<option value=\"0\">Selecciona un autor/a </option>";  
40  
41     foreach($autors as $autor) {  
42         $resultat .= "<option value=" . $autor->id . ">" . $autor->Cognom1 . " " . $autor->Cognom2 . ", " .  
43             $autor->Nom . "</option>";  
44     }  
45  
46     return $resultat;  
47 }
```

Elaboración propia

La función del controlador básicamente lo que hace es buscar todos aquellos autores/as cuyo primer apellido empiece por la letra seleccionada. Luego, me guardo en la variable *\$resultat* el resultado, que en la captura anterior se inyecta al componente con la función *innerHTML*. El *foreach* lo que hace es recorrer todos los autores/as obtenidos en la primera consulta y los va acumulando en la variable *\$resultat*.

Figura 6.7: Resultado de aplicar el código de la Figura 6.4, Figura 6.5 y Figura 6.6

The image shows a web interface with a dark grey top navigation bar. The first item on the left is 'ÍNDIX AUTORS/ES' in white text on a red background. The other items are 'CERCADOR AUTORS/ES', 'OBRES', 'CERCADOR OBRES', and 'ALTRES CONSULTES', each with a small icon above the text. Below the navigation bar is a large white area. At the top of this area is a grid of 26 buttons representing the alphabet, arranged in two rows of 13. The button for 'E' is highlighted in blue. Below the grid is a search input field with the placeholder text 'Selecciona un autor/a' and a dropdown arrow on the right. At the bottom of the white area is a green button with a magnifying glass icon and the text 'Buscar'.

Elaboración propia

Nota: Al igual que en el caso de los diagramas, no he podido respetar los márgenes interior y exterior para su correcta visualización.

6.1.1 Errores

Se controla que el usuario no pueda realizar una búsqueda sin haber seleccionado una opción del abecedario o si intenta buscar por la opción por defecto (Selecciona un autor/a), como se puede ver a continuación:

Figura 6.8: El usuario busca sin seleccionar una letra o autor/a


The image shows a web interface for searching authors and books. At the top, a pink warning banner reads "Has de seleccionar algun autor" with a close button. Below this is a navigation bar with five items: "ÍNDIX AUTORS/ES" (highlighted in pink), "CERCADOR AUTORS/ES", "OBRES", "CERCADOR OBRES", and "ALTRES CONSULTES". The main content area contains a grid of 26 buttons representing the alphabet (A-Z). Below the grid is a dropdown menu with the text "Selecciona una lletra del abecedari". At the bottom is a large green button with a magnifying glass icon and the text "Buscar".

Elaboración propia

6.2 Resultado del proyecto VS *Mockups*

Tras finalizar la implementación, los resultados finales de las páginas principales en comparación a cómo lo tenía en los *mockups*, han quedado de la siguiente forma:

Figura 6.9: Página principal con el primer tab: Índex Autors/es

 Actualització de dades

Lletraferit

Contacta'ns: biel@ua.es

ÍNDEX AUTORS/ES

CERCADOR AUTORS/ES

OBRES

CERCADOR OBRES

ALTRES CONSULTES

A B C D E F G H I J K L M


N O P Q R S T U V W X Y Z

Selecciona una lletra del abecedari

Buscar

Elaboración propia

Figura 6.10: Página principal con el segundo tab: Cercador Autors/es

 Actualització de dades

Lletraferit

Contacta'ns: biel@ua.es



ÍNDIX AUTORS/ES

CERCADOR AUTORS/ES

OBRES


CERCADOR OBRES

ALTRES CONSULTES

☐  ☐ 

Elaboración propia

Figura 6.11: Página principal con el tercer tab: Obres

 Actualització de dades

Lletraferit

Contacta'ns: biel@ua.es

ÍNDIX AUTORS/ES

CERCADOR AUTORS/ES

OBRES

CERCADOR OBRES

ALTRES CONSULTES

A B C D E F G H I J K L M

N O P Q R S T U V W X Y Z

Elaboración propia

Figura 6.12: Página principal con el cuarto tab: Cercador Obres

 Actualització de dades

Lletraferit

Contacta'ns: biel@ua.es

 ÍNDEX AUTORS/ES

 CERCADOR AUTORS/ES

 OBRES

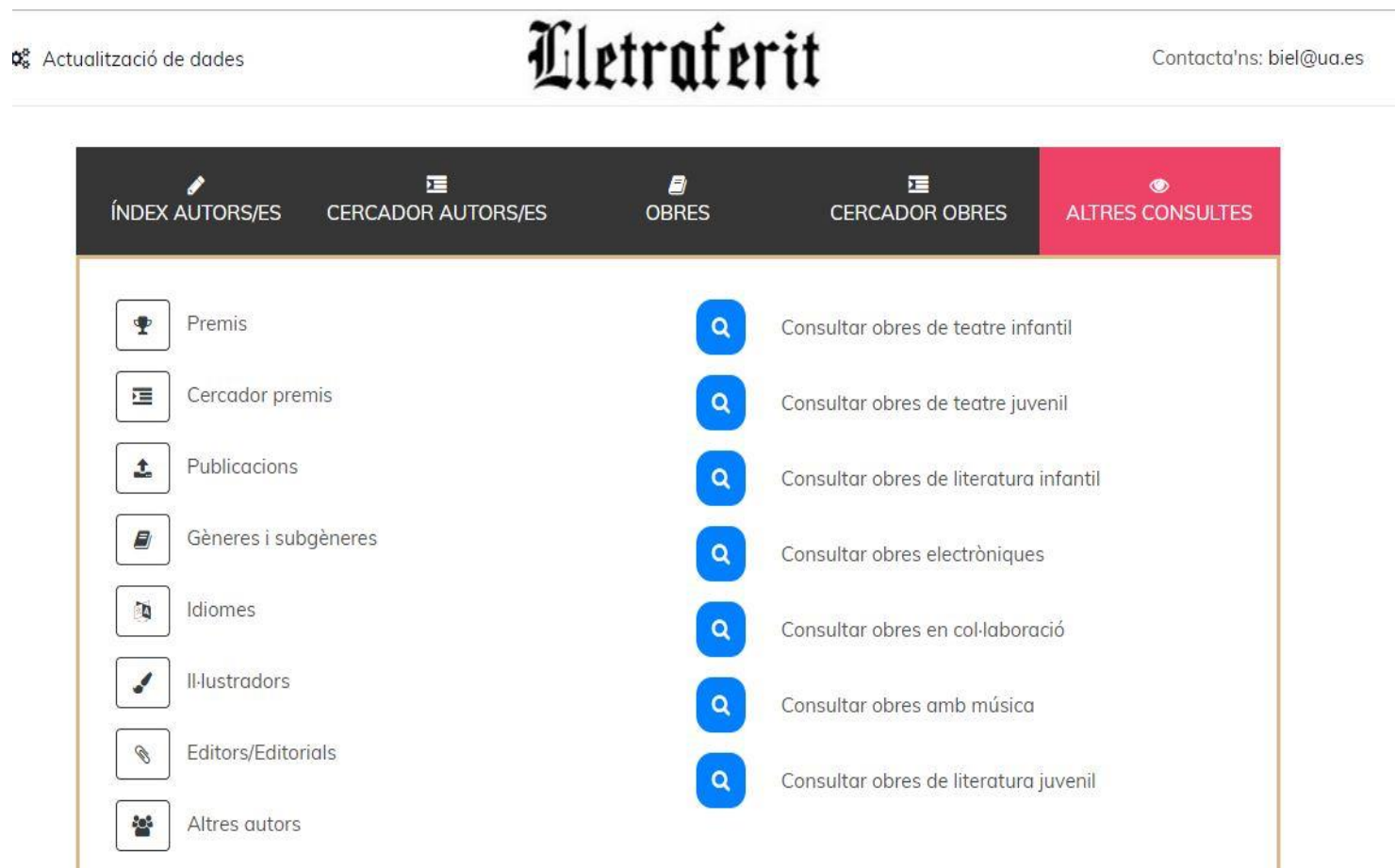
 CERCADOR OBRES

 ALTRES CONSULTES

 Buscar

Elaboración propia

Figura 6.13: Pàgina principal con el quinto tab: Altres Consultes



Elaboración propia

Figura 6.14: Pàgina de informació del autor

⚙ Actualització de dades


Llettraferit

Contacta'ns: biel@ua.es

←

Ortega Rodríguez, Vicent

Autor




Informació	
Nom: Vicent	Comarca de naiximent: la Ribera del Xúquer
Cognom1: Ortega	Data de naiximent: 1957-10-02 00:00:00
Cognom2: Rodríguez	Professió: Professor d'EPA
Pseudonim: Vicent Ortega	Sexe: M
Lloc de naiximent: L'Alcúdia	
Lloc de residència: L'Alcúdia	


Formació

Vicent Ortega Rodríguez va néixer a l'Alcúdia (la Ribera del Xúquer) el 1957. És llicenciat en Filologia Castellana i Francesa i diplomad en Magisteri per la Universitat de València. Actualment exerceix com a professor de primària. La seua faceta com a creador s'ha reflectit en la publicació de dues obres: un poemari, *Mediterrània* (1995) i les novel·les, *El crim* (1996), finalista del Premi de Novel·la Ciutat d'Alzira 1996 i del Premi Ramon Llull de les Lletres Catalanes 1997 o *El collidor de ànimes* (2001).


Obres




Collidor d'ànimes, El



Crim, El




Diumenge de glòria




Mediterrània


Elaboración propia

Figura 6.15: Pàgina de informació de una obra


 Actualització de dades



Contacta'ns: biel@ua.es




Crim, El



Informació
Títol: Crim, El
Subtítol: No disponible
Gènere: Novel·la
Subgènere: No disponible
Pròleg: No disponible
Introducció: No disponible
Epíleg: No disponible
Il·lustrador: No disponible

Autor@/s que han escrit l'obra

Autor	Informació
Ortega Rodríguez, Vicent	

Elaboración propia

Figura 6.16: Pàgina de informació de un premio

 Actualització de dades



Contacta'ns: biel@ua.es



Narcís Oller

Descripció

Atorgat per la Diputació de Tarragona

Obres que han guanyat el premi



Contra sentit

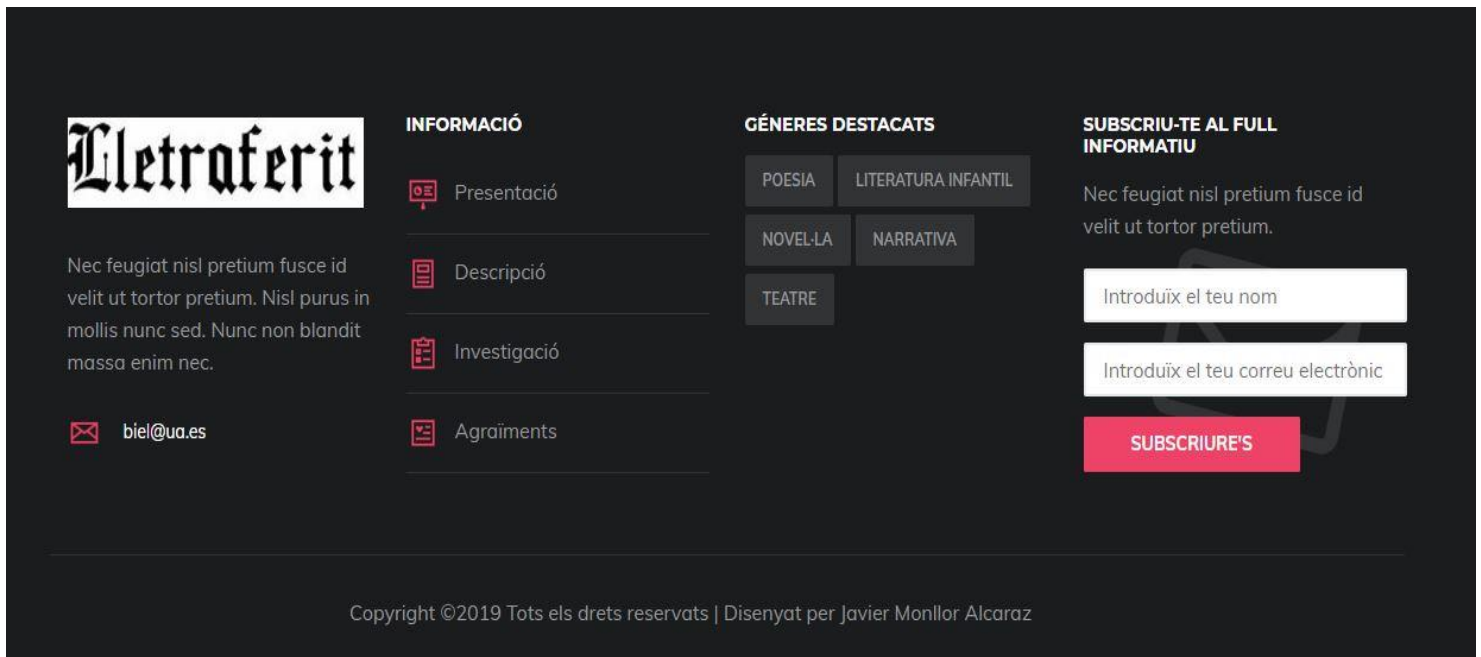
Elaboración propia

Figura 6.17: Pàgina de informació de un idioma

Bandera	Idioma	Descripció	Consultar obres
	alemany		
	anglès		
	castellà		
	eslovac		

Elaboración propia

Figura 6.18: Footer



Elaboración propia

Nota: Por la misma razón comentada anteriormente, no he podido respetar los márgenes interior y exterior para su correcta visualización.

Como se puede observar, se han aplicado muchos cambios con respecto a los *mockups* iniciales. Entre ellos destacar los siguientes:

- 5 opciones en lugar de 4:
 - Índex autors/es
 - Cercador autors/es
 - Obres
 - Cercador obres
 - Altres consultes
- El menú horizontal con las cuatro opciones se ha eliminado y reestructurado en el *footer* en el apartado “Información”.

- Página de información del autor eliminada y unida en Página de resultados de búsqueda.
- Página de información de un premio se muestra a modo de tabla, al igual que la de idiomas.
- Eliminación de las Redes Sociales.
- Modificación de dos apartados en el *Footer*:
 - Contactar -> Informació
 - Tags -> Gèneres destacats
- Introducción de un abecedario para las búsquedas individuales.
- Al buscar una obra buscamos por:
 - Título
 - Género
- Al buscar un autor buscamos por:
 - Nombre o Pseudónimo
 - Lugar de nacimiento
 - Comarca de nacimiento
 - Sexo

6.3 Implementar el *back office*

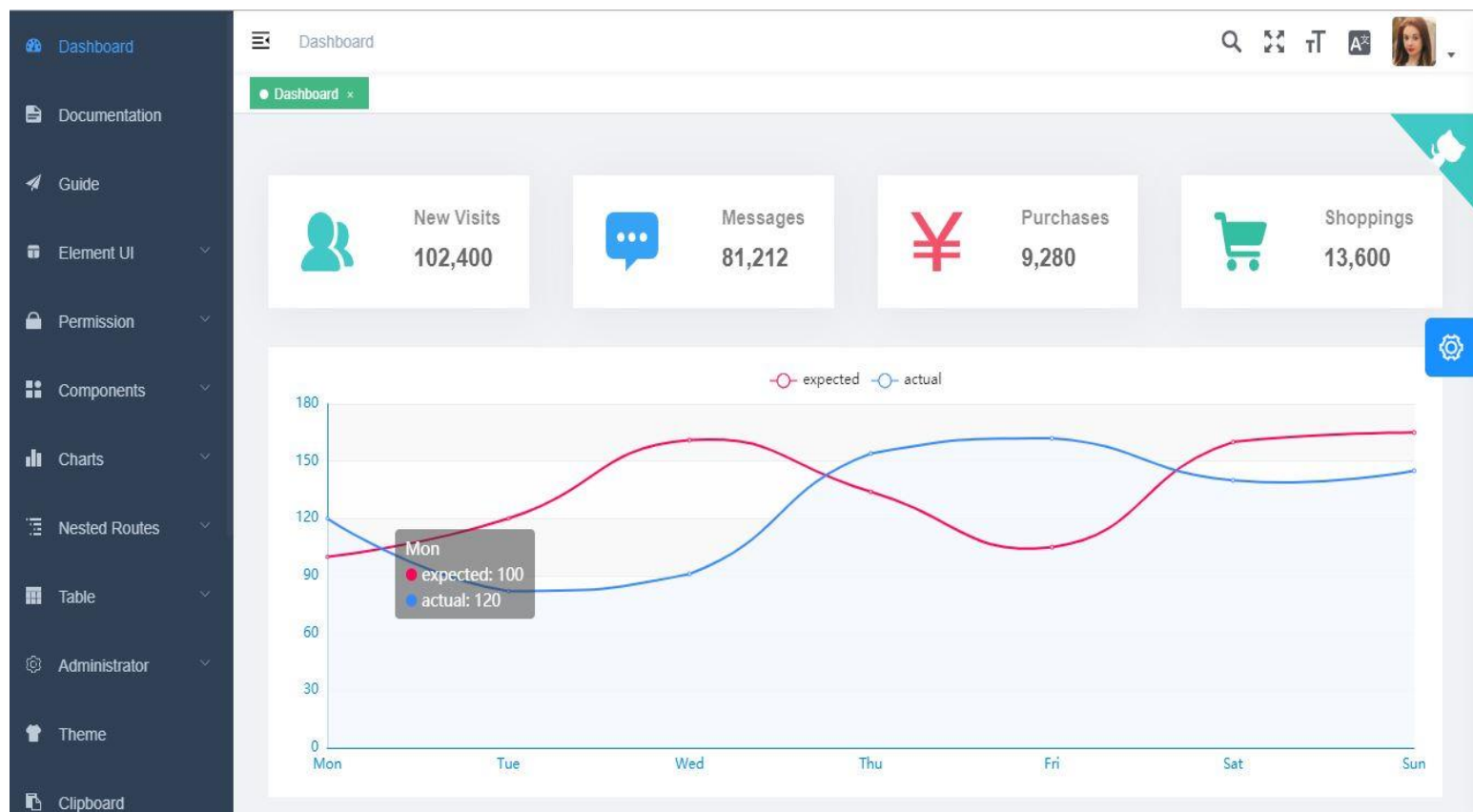
Cuando el *front office* esté implementado al 100%, se creará y subirá a Github otro proyecto que contendrá una web de *back office* o de trastienda que permita a los usuarios mantener los datos.

Por lo que respecta a la parte del *back office*, he usado el Laravue que me hizo aumentar la productividad. [3] Además integre BootstrapVue con la idea de usar sus componentes en el diseño de las vistas. En ese sentido he utilizado los siguientes componentes:

- `<b-table>`
 - Implementar la tabla con los valores, pudiendo realizar una ordenación por columnas y un filtro.
- `<b-form-select>`
 - Mostrar el *select* con la opción de mantener el nombre y almacenar el id.
- `<b-input-group>`
 - Input para poder introducir texto.

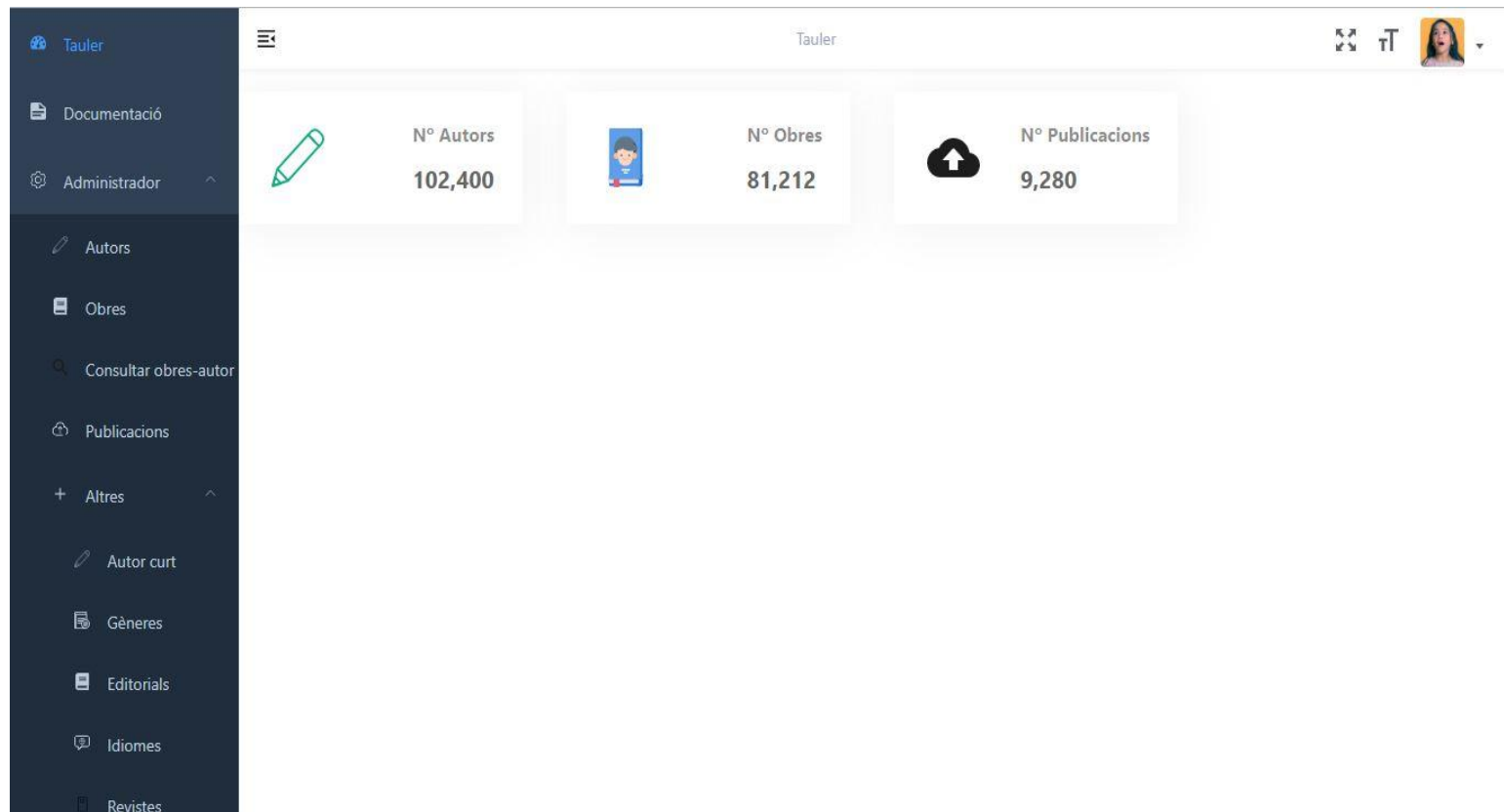
Sobre el proyecto original he realizado una gran cantidad de cambios, sobre todo en el panel administrativo quedando de la siguiente forma:

Figura 6.19: Aspecto de back office inicial



Elaboración propia

Figura 6.20: Aspecto del back office final



Elaboración propia

Nota: No he podido respetar los márgenes interior y exterior para su correcta visualización.

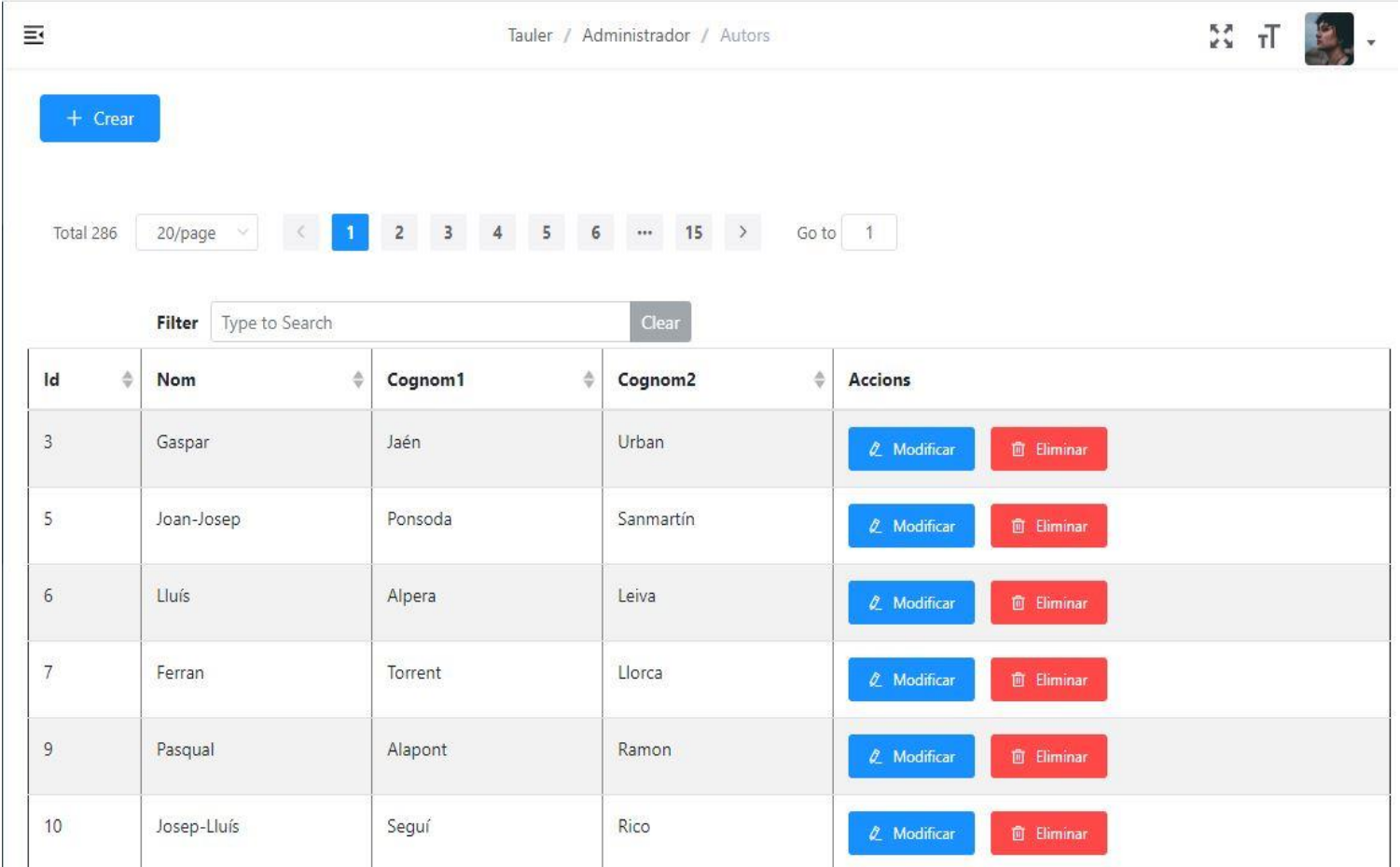
[1] El *back office* será el lugar desde donde el *webmaster* o administrador mantendrá y actualizará los datos. He tenido en cuenta patrones de diseño web que faciliten el uso a los administradores, como por ejemplo:

- Paginación.
- Ordenación.
- Formularios inteligentes.
- Preguntas de confirmación antes de borrar un elemento.

- Búsqueda por nombre.
- Identificación de usuarios.
- Pantalla completa.

Para ello, he implementado los siguientes *CRUDs*:

Figura 6.21: Listado de todos los autores



Id	Nom	Cognom1	Cognom2	Accions
3	Gaspar	Jaén	Urban	Modificar Eliminar
5	Joan-Josep	Ponsoda	Sanmartín	Modificar Eliminar
6	Lluís	Alpera	Leiva	Modificar Eliminar
7	Ferran	Torrent	Llorca	Modificar Eliminar
9	Pasqual	Alapont	Ramon	Modificar Eliminar
10	Josep-Lluís	Seguí	Rico	Modificar Eliminar

17	Josep	Piera	Rubio	Modificar	Eliminar
18	Carles	Cano	Peiró	Modificar	Eliminar
19	Vicent	Alonso	Catalina	Modificar	Eliminar
20	Josep-Lluís	Sirera	Turó	Modificar	Eliminar
21	Rodolf	Sirera	Turó	Modificar	Eliminar
22	Emili	Rodríguez	Bernabeu	Modificar	Eliminar
23	Àlan	Greus	Bosch	Modificar	Eliminar
24	Diego	Rodríguez	Herrero	Modificar	Eliminar
25	Vicent	Pardo	Peris	Modificar	Eliminar

Total 286

20/page ▾

<

1

2

3

4

5

6

...

15

>

Go to

1

Elaboración propia

Figura 6.22: Añadir un nuevo autor al pulsar el botón Crear

Crear un nou autor

Nom

Primer cognom

Segon cognom

Pseudonim

I

No

Lloc de naiximent

Lloc de residència

Comarca de naixime

nt

Data de naiximent	<input type="text" value="dd/mm/aaaa"/>
Formació	<input type="text"/>
Observacions	<input type="text"/>
Professió	<input type="text" value=""/>
Sexe	<input type="text" value=""/>
Foto perfil	<div><div>Seleccionar archivo</div><div>Ningún archivo seleccionado</div></div>
Blog	<input type="text"/>
<div><div>Cancel·lar</div><div>Confirmar</div></div>	

Elaboración propia

Figura 6.23: Modificar un autor al pulsar el botón Modificar

Editar autor

Nom

Gaspar

Primer cognom

Jaén

Segon cognom

Urban

Pseudonim

Gaspar Jaén

I

Sí

Lloc de naiximent

Elx

Lloc de residència

Elx

Comarca de naiximent


el Baix Vinalopó

Data de naiximent	<input type="text" value="dd/mm/aaaa"/>
Formació	<input type="text" value="Gaspar Jaén Urban va nèixer a Elx (el Baix Vinalo pó) el 1952. Es va titular en l'Escola Superior d'Ar"/>
Observacions	<input type="text"/>
Professió	<input type="text" value="Professor universitari"/>
Sexe	<input type="text" value="Masculí"/>
Foto perfil	<input type="button" value="Seleccionar archivo"/> <input type="text" value="Ningún archivo seleccionado"/>
Blog	<input type="text"/>
	<input type="button" value="Cancel·lar"/> <input type="button" value="Confirmar"/>

Elaboración propia

Figura 6.24: Eliminar un autor al pulsar sobre el botón Eliminar

Advertència



Açò eliminarà permanentment l'autor Gaspar.
Continuar?

Elaboración propia

Figura 6.25: Listado de todas las obras

Tauler / Administrador / Obres

T

+ Crear

Total 2171

20/page

<

1

2

3

4

5

6

...

109

>

Go to

1

Filter

Type to Search

Clear

Id	Titol	Subtitol	Observacions	Accions
2	Cadells de la fosca trencada			<div>Modificar</div> <div>Eliminar</div>
3	Cambra de mapes			<div>Modificar</div> <div>Eliminar</div>
4	Festa, La			<div>Modificar</div> <div>Eliminar</div>
5	Fragments			<div>Modificar</div>
21	Surant enmig del naufragi final contemple el voluptuós incendi de totes i cadascuna de les flors del núbil hibiscus			<div>Modificar</div> <div>Eliminar</div>
22	Tempesta d'argent			<div>Modificar</div> <div>Eliminar</div>
23	Esplendor de l'àmfora, L'			<div>Modificar</div> <div>Eliminar</div>
24	Ciudad irisada de las buguenvillas / Ciutat irisada de les buguenvil·lees, la (antologia bilingüe)			<div>Modificar</div> <div>Eliminar</div>
25	Fidels dits del cant, Els			<div>Modificar</div> <div>Eliminar</div>
26	Dons del pleniluni, Els			<div>Modificar</div> <div>Eliminar</div>

Total 2171

20/page

<

1

2

3

4

5

6

...

109

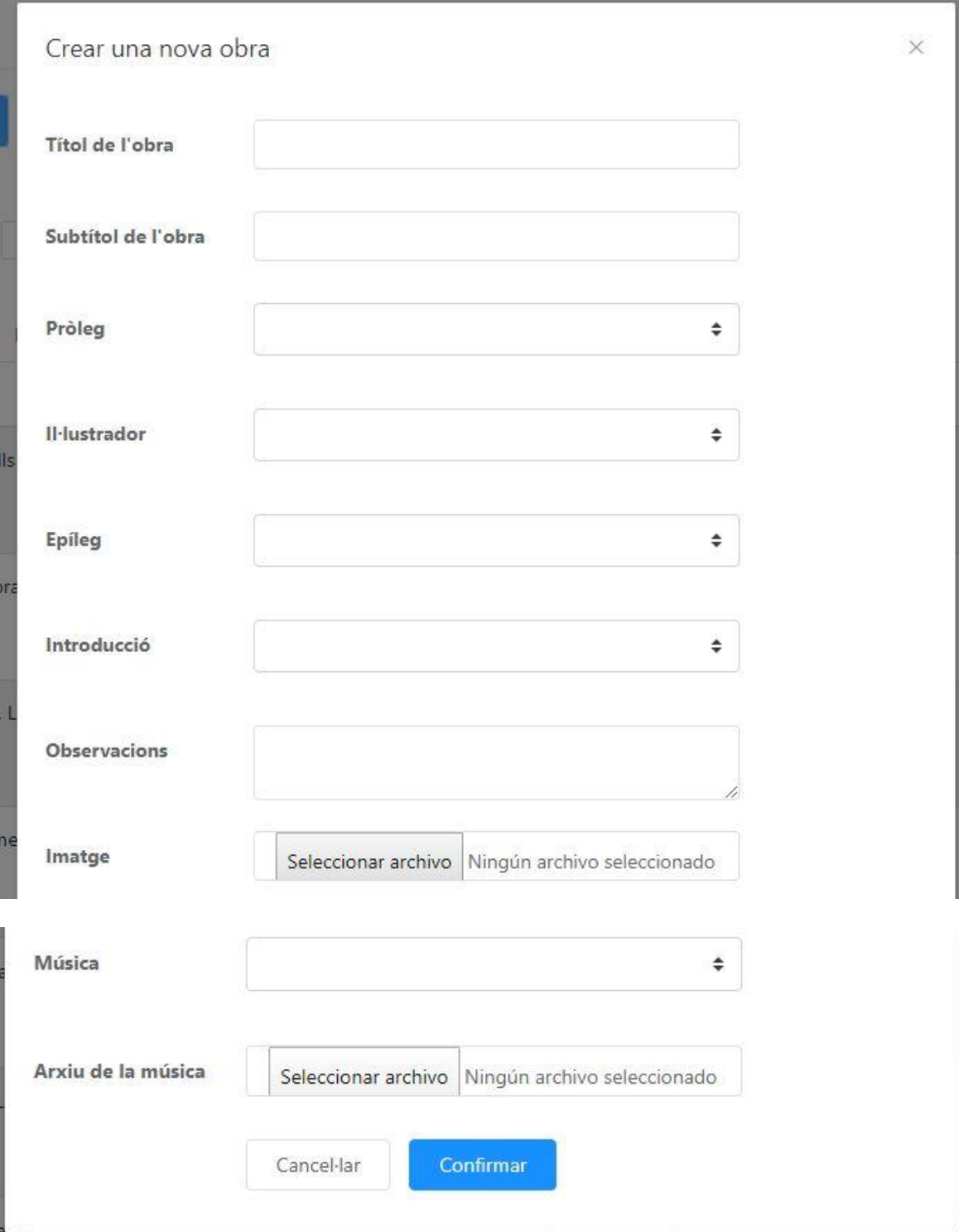
>

Go to

1

Elaboración propia

Figura 6.26: Añadir una nueva obra al pulsar el botón Crear



Crear una nova obra ✕

Títol de l'obra

Subtítol de l'obra

Pròleg

Il·lustrador

Epíleg

Introducció

Observacions

Imatge Ningún archivo seleccionado

Música

Arxiu de la música Ningún archivo seleccionado

Elaboración propia

Figura 6.27: Modificar una obra al pulsar el botón Modificar

Editar obra

Títol de l'obra

Taula dels poetes de l'illa coronada, La

Subtítol de l'obra

Pròleg

Il·lustrador

Epíleg

Introducció

Observacions

(Antologia).

Imatge

Seleccionar archivo

Ningún archivo seleccionado

Música

Sí

Arxiu de la música

Seleccionar archivo

Ningún archivo seleccionado

Cancel·lar

Confirmar

Elaboración propia

Figura 6.28: Eliminar una obra al pulsar sobre el botón Eliminar



Elaboración propia

Figura 6.29: Listado de todas las publicaciones

Tauler / Administrador / Publicacions

T

+ Crear

Total 2411

20/page

<

1

2

3

4

5

6

...

121

>

Go to

1

Filter

Type to Search

Clear

Id	Any	Col·leccio	ISBN DL	Accions
1	1976		84852120-0 848521120-0	<div>Modificar</div> <div>Eliminar</div>
2	1982	Llibres del Mall	84-7456 108 6	<div>Modificar</div> <div>Eliminar</div>
3	1983	Balenguera, col·lecció de poesia	84-273 0420 X	<div>Modificar</div> <div>Eliminar</div>
4	1991	Tres i quatre/poesia		<div>Modificar</div> <div>Eliminar</div>
5	1998	Sinera		<div>Modificar</div> <div>Eliminar</div>
7	1984	Papers d'Elx	84-86114 06 3	<div>Modificar</div> <div>Eliminar</div>

14	1963			Modificar	Eliminar
15	1967			Modificar	Eliminar
16	1968			Modificar	Eliminar
17	1971			Modificar	Eliminar
18	1980			Modificar	Eliminar
19	1985			Modificar	Eliminar
20	1986	Els llibres de l'Escorpí. Poesia	84-297 2488 5	Modificar	Eliminar
21	1987		84-398 9790 1	Modificar	Eliminar
22	1987			Modificar	Eliminar

Total 2411

20/page ▾

<

1

2

3

4

5

6

...

121

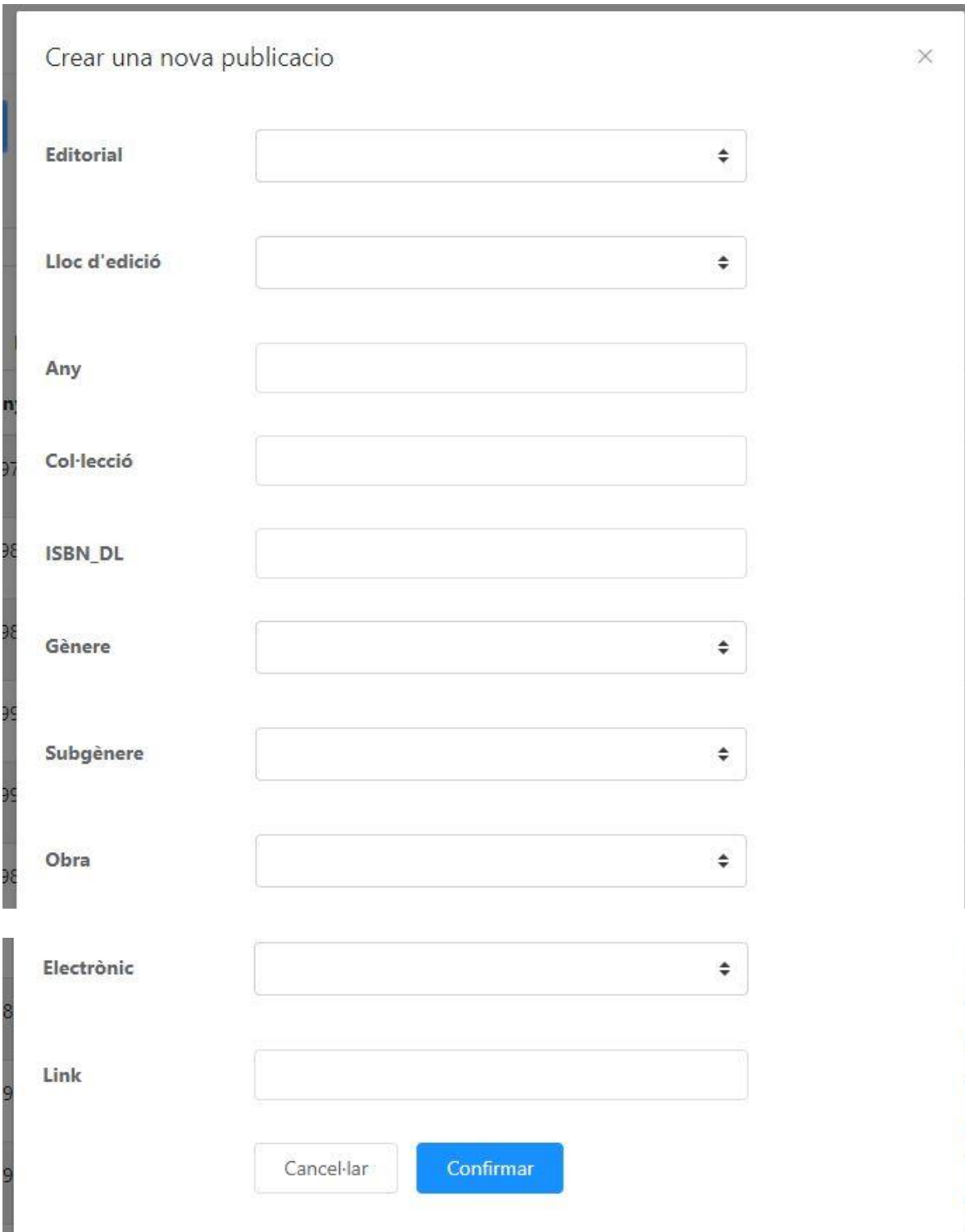
>

Go to

1

Elaboración propia

Figura 6.30: Añadir una nueva publicación al pulsar el botón Crear



Crear una nova publicacio ×

Editorial

Lloc d'edició

Any

Col·lecció

ISBN_DL

Gènere

Subgènere

Obra

Electrònic

Link

Elaboración propia

Figura 6.31: Modificar una publicación al pulsar el botón Modificar

Editar publicacio

Editorial

Eliseu Climent

Lloc d'edició

València

Any

1976

Col·lecció

ISBN_DL

84852120-0848521120-0

Gènere

Poesia

Subgènere

Obra

Cadells de la fosca trencada

Electrònic

Sí

Link

<https://www.google.com/>

Cancel·lar

Confirmar

Elaboración propia

Figura 6.32: Eliminar una publicación al pulsar sobre el botón Eliminar



Elaboración propia

Nota: No he podido respetar los márgenes interior y exterior para su correcta visualización.

Por último, hay una opción llamada Altres que agrupa el mantenimiento de varios CRUD como en los casos anteriores:

- Autor curt
- Gèneres
- Editorials
- Idiomes
- Revistes
- Crítiques
- Premis

Capítulo 7

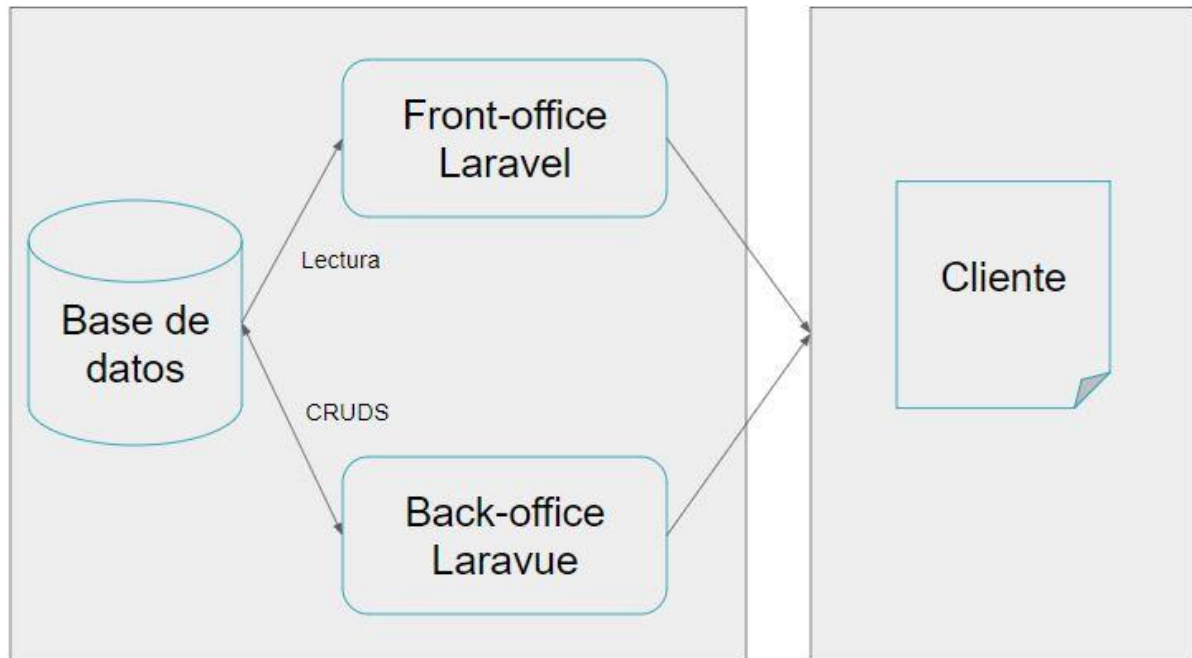
Puesta en producción

Una vez finalizamos la implementación y pruebas, tuvimos que poner el sistema en producción, primero durante un periodo de prueba para que el administrador vea como ha quedado en real y proponga los últimos cambios o correcciones.

Para subirlo a producción, tuvimos que hablar con el administrador del servidor web donde lo alojaremos para solicitarle acceso FTP y SSH con el fin de configurar correctamente el *framework* Laravel y demás tecnologías usadas, dotar de los permisos necesarios a ciertos directorios y ficheros...

El diagrama de arquitectura del proyecto ha quedado de la siguiente forma:

Figura 7.1: Esquema del proyecto



Elaboración propia

Tras este periodo, que en el momento de la redacción de este documento, no ha finalizado, tendremos que establecer una fecha para la salida definitiva a real. Para ello, limpiaremos datos de prueba, acabaremos de depurar la base de datos, solucionaremos los errores de referencias e incoherencias de datos encontrados, referidos en el capítulo 9 (según lo acordado con Biel y su equipo) y ya tendremos la web publicada definitivamente.

Capítulo 8

Pruebas del sistema

A lo largo del desarrollo del proyecto, se han ido realizando una serie de pruebas para comprobar que las funciones o métodos se estaban implementando de forma correcta. Las pruebas utilizadas en este caso han sido las siguientes:

- Pruebas unitarias
- Pruebas de funcionalidad
- Pruebas de interfaz/usabilidad

8.1 Pruebas unitarias

Las pruebas unitarias consisten en aislar una parte del código y comprobar que funciona perfectamente con el paso de parámetros. En este caso, se invocan las rutas directamente con el paso de parámetros y se verifican los resultados con la función *assert*. Se han realizado varias pruebas, tanto en la parte del *front-end* como en la de *back-end*. Entre ellas destacaría las siguientes.

8.1.1 *Fron-end*

Una de las pruebas realizadas ha sido llamar a la ruta */indexObres* con los siguientes parámetros:

- Títol de l'obra = **obra**
- Gènere = **NULL**

Como resultado he obtenido un *array* de obras, pero me he guardado el número de obras obtenidas. Como resultado esperado tenía que obtener un total de 21 obras. Al invocar a la función *assert* con el resultado obtenido y resultado esperado me ha devuelto *true*, por tanto esa parte del código funciona perfectamente.

8.2 Pruebas de funcionalidad

Las pruebas funciones consisten en comprobar que las funcionalidades del software hacen todo lo que deberían hacer y de forma correcta. Al igual que en las pruebas anteriores, se han realizado varias pruebas en *back-end* y *front-end*.

8.2.1 *Front-end*

En el apartado “Subscriu-te al full informatiu” del *footer* el usuario introduce los siguientes datos:

- Nom = **Javier**

- Correu electrònic = jma86@alu.ua.es

Como resultado, aparece un mensaje de confirmación de la suscripción y los datos se han almacenado en la tabla “subscribers”.

8.2.2 *Back-end*

El administrador en el *back office* selecciona la opción Autores del panel lateral y realiza las siguientes acciones:

- Crear autor:
 - Nom = **Javier**
 - Primer Cognom = **Monllor**
 - Segon Cognom = **Alcaraz**
 - Lloc de naiximent = **Tibi**
 - Comarca de naiximent = **Alacant**
 - Sexe = **M**

Como resultado, se crea un nuevo autor con los campos introducidos por el administrador y el resto NULL en la tabla *eauteurs*. Al listar los autores ya aparece el nuevo introducido.

- Modificar autor:
 - Nom = **JavierModificado**
 - Data de naiximent = **29/01/1997**

Como resultado, se guardan los cambios realizados sobre el mismo autor en la tabla *eauteurs*. Al listar los autores ya aparece el autor modificado.

- Eliminar autor

Al pulsar sobre el botón *Eliminar* de un autor en concreto y confirmar el borrado, el autor se elimina de la tabla *eauteurs*. Al listar los autores ya no aparece el autor eliminado.

8.3 Pruebas de interfaz/usabilidad

Las pruebas de interfaz/usabilidad consisten en evaluar el sitio web con la ayuda de un grupo de usuarios. Estos usuarios van a interactuar con el sitio web comprobando que se cumpla el propósito para el cual fue diseñado, tomando nota de aquellos errores o problemas que han tenido para realizar cierta tarea. He utilizado como usuarios reales a miembros de mi familia. En este caso, no he realizado tantas pruebas como en los casos anteriores, entre ellas destacaría las siguientes.

8.3.1 *Front-end*

El objetivo del usuario en el *front office* era consultar todas aquellas obras cuyo género fuera la Poesía. Como resultados obtuve lo siguiente:

- Tiempo empleado = **1:34**
- Dificultad (1-6) = **3**
- Valoración = **Normal**

- Observaciones = **Al ser mi primera vez me ha costado un poco encontrarlo, ya que se encuentra en la quinta pestaña y el nombre de “Gèneres i subgèneres” pensaba que era para otra cosa.**

8.3.2 Back-end

El objetivo del administrador en el *back office* era crear una nueva obra. Como resultados obtuve lo siguiente:

- Tiempo empleado = **0:45**
- Dificultad (1-5) = **1**
- Valoración = **Muy fácil**
- Observaciones = **Aunque haya sido mi primera vez, no me ha costado nada encontrar en el panel la opción de obras y ver el botón de crear. Al rellenar los campos tampoco he tenido problemas.**

Capítulo 9

Problemas y soluciones

A lo largo de la implementación del sitio web, tuve muchos problemas sobre todo al inicio del proyecto. Algunos problemas y soluciones que aporté han sido los siguientes.

9.1 Base de datos

La base de datos que tenía que reutilizar del anterior sitio web era en formato Access, lo cual me obligó a realizar una conversión a MySQL que era con lo que iba a trabajar. Cuando realice la primera conversión, utilice un sitio online que me permitía realizar la conversión sin problema. Pero, al importar el resultado de la conversión a WorkBench para poder observar las tablas y datos, me di cuenta de las siguientes cosas:

- Falta de datos en algunas tablas.
- El *Datatype* de algunas columnas eran incorrectos:
 - Presencia de *Varchar(0)*.
- Las columnas no estaban identificadas:
 - No estaban marcadas las claves primarias.
 - No estaban marcadas las claves ajenas.
 - No sabía qué columnas no podían ser nulas.
 - Todas las columnas por defecto eran nulas.

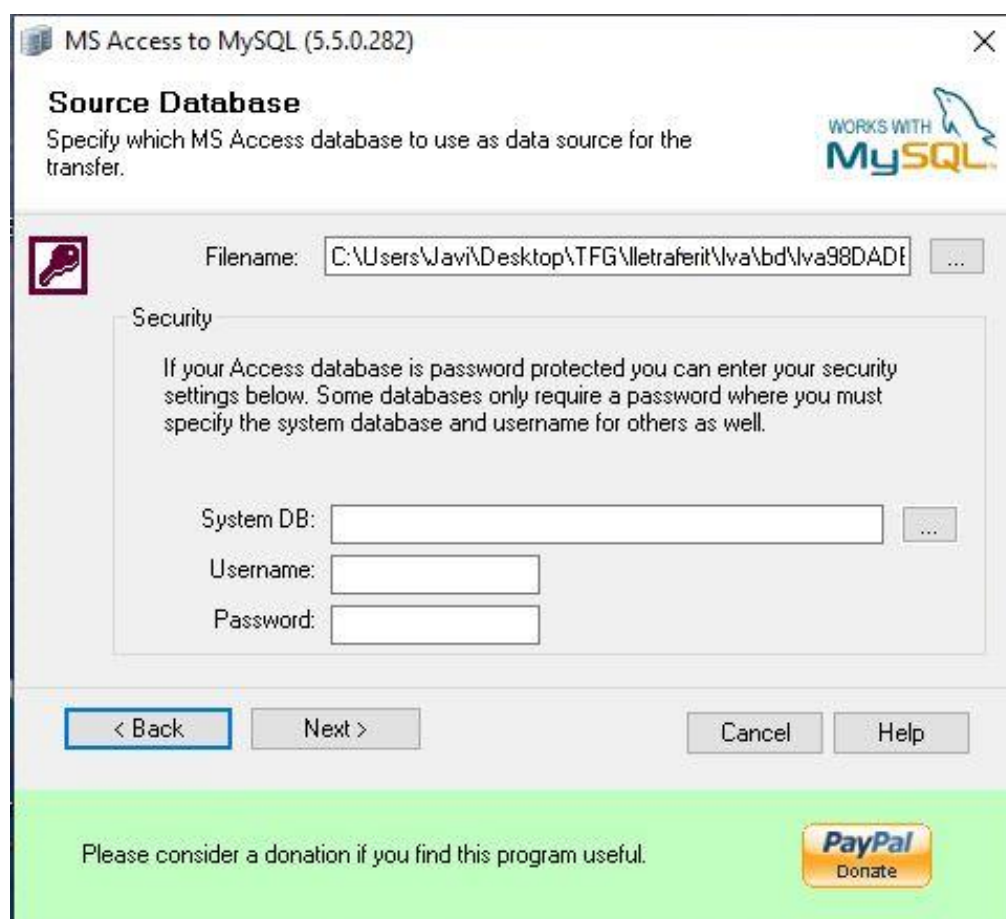
- Presencia de en tablas de N:M valores 0 que no relacionan a nadie.
- Presencia de en tablas de 1:N valores 0 que no relacionan a nadie.

9.1.1 Solución

Como solución a los problemas comentados anteriormente, encontré un programa que también realizaba la conversión pero parecía mucho más fiable que en el caso del online. El programa como he comentado en el apartado de Tecnologías, se llama Bullzip MS Access to MySQL, y los pasos para la conversión son muy fáciles y son los siguientes:

1. Si la base de datos de Access requiere que inicie sesión, tendrás que especificar el nombre de usuario, contraseña y una base de datos del sistema. En mi caso no he necesitado realizar ninguna identificación.

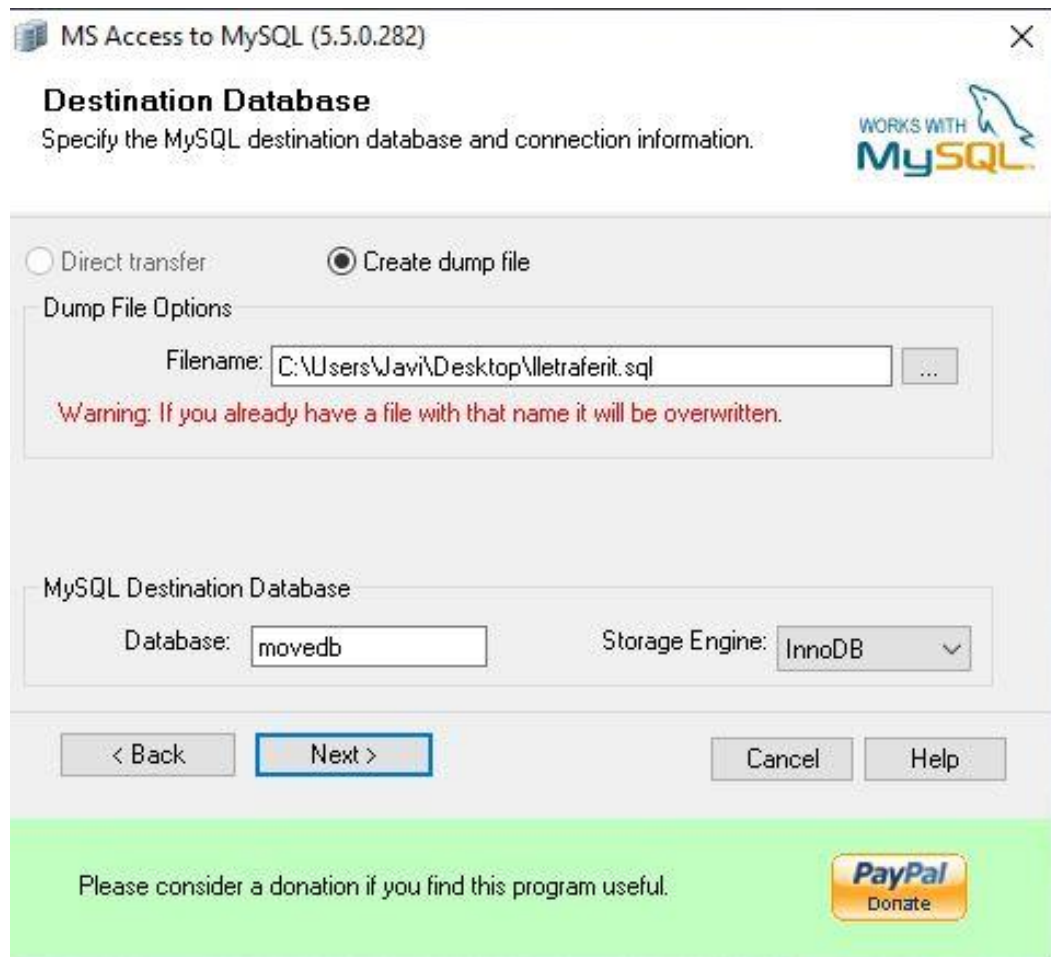
Figura 9.1: Localizar la base de datos a convertir e identificación si es necesario



Elaboración propia

2. A continuación, introducimos donde se va a generar el esquema y con qué nombre, el nombre de la base de datos y *Storage Engine*.

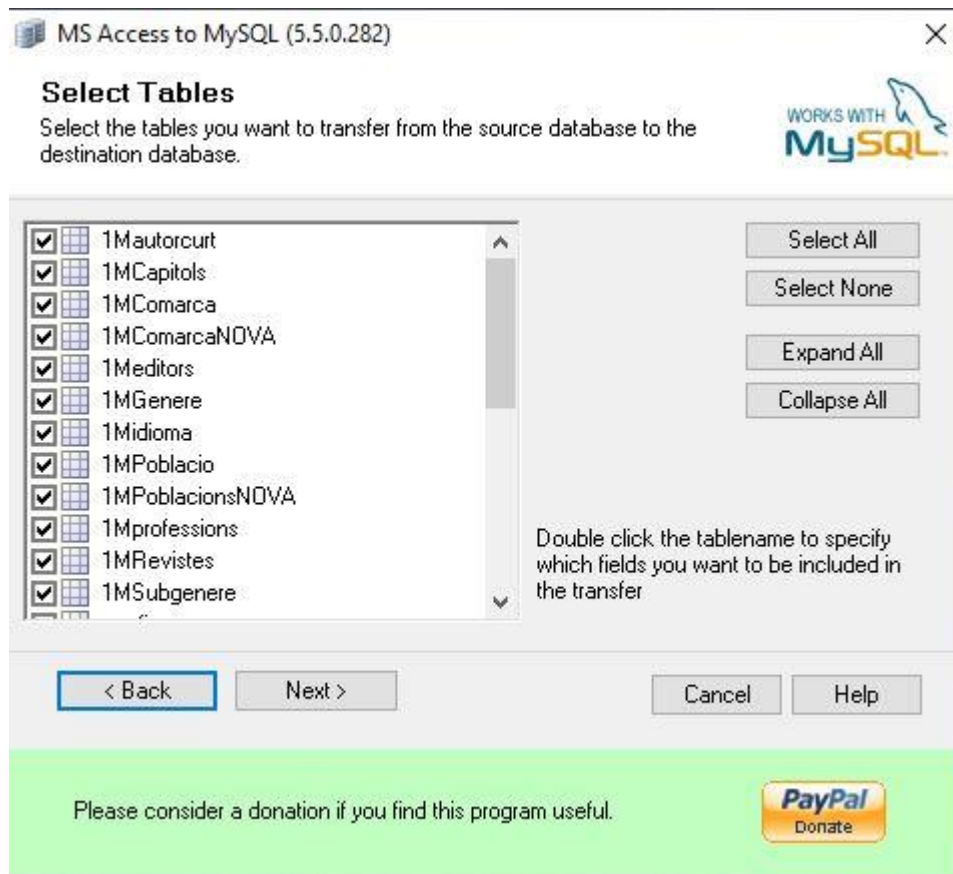
Figura 9.2: Nombre de la nueva base de datos y localización del fichero



Elaboración propia

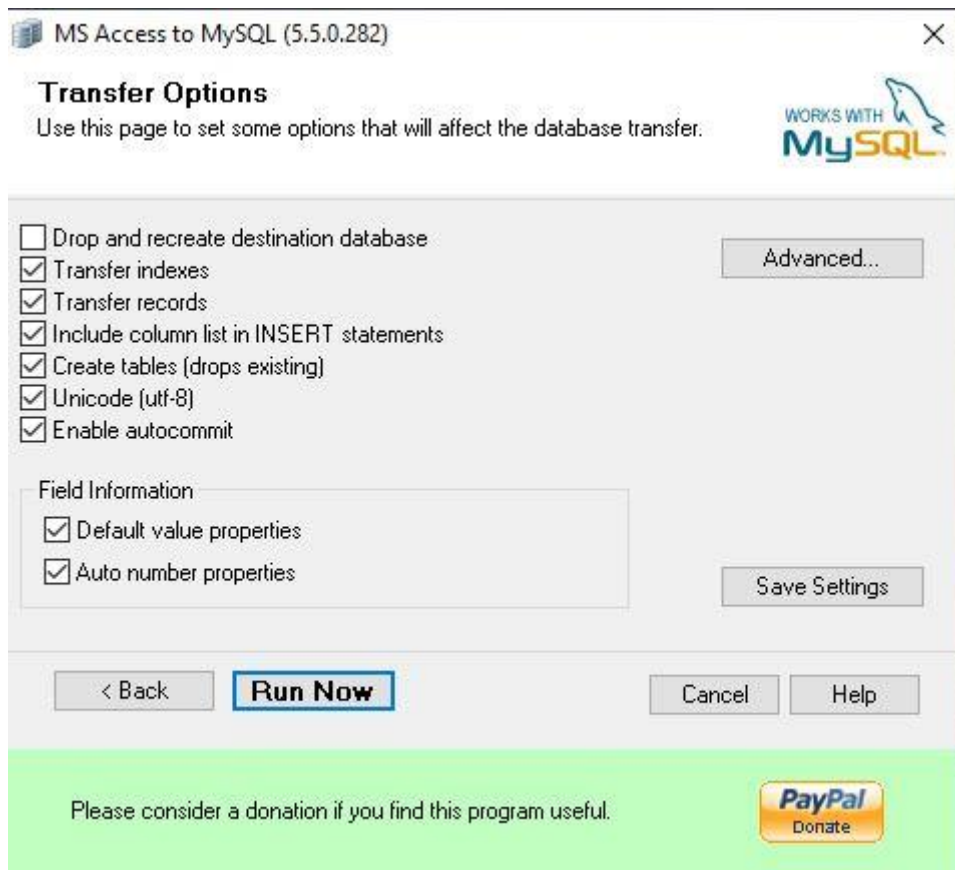
3. Por último, seleccionamos todas las tablas que queramos convertir, las opciones de transferencia y ejecutamos.

Figura 9.3: Tablas a convertir



Elaboración propia

Figura 9.4: Configuraciones de la nueva base de datos



Elaboración propia

Utilizando esta opción, he conseguido solucionar algunos de los problemas que tenía en la opción del online que son los siguientes:

- + Presencia de todos los datos.
- + El *Datatype* de todas las columnas es correcto.
- + Presencia en cada tabla de su clave primaria autoincremental.
- + Presencia en cada tabla de las columnas que no pueden ser nulas.
- + Por defecto todas las columnas no son nulas.

No todo son ventajas utilizando este programa, me he dado cuenta que se mantienen algunos problemas comentados anteriormente más algún problema adicional que son los siguientes:

- No hay ninguna clave ajena marcada.
- Presencia de en tablas de N:M valores 0 que no relacionan a nadie.
- Presencia de en tablas de 1:N valores 0 que no relacionan a nadie.
- Las columnas que no pueden ser nulas tienen un valor por defecto 0.
- Columnas con valores 0 que no relacionan a otra tabla.

En conclusión, esta herramienta me ha ahorrado mucho trabajo, pero los nuevos problemas que han surgido más lo que no ha conseguido solucionar del primer caso los he tenido que solucionar a mano. He tenido que relacionar todas las tablas, lo cual no ha sido pesado ya que podía observar en el proyecto antiguo como estaban relacionadas cuando tenía alguna duda. Las columnas con valores 0 que relacionan a otra tabla, he decidido eliminar toda la fila asociada. Las columnas con valores 0 que no relacionan a otra tabla, he decidido mantener la fila pero modificando el valor 0 del campo por NULL. Todas las columnas que no podían ser nulas, he eliminado su valor por defecto, ya que algunas tenían como valor por defecto 0 y otras NULL, cosa que no tiene sentido.

9.2 Integración de Laravue

En un primer momento, clone el repositorio que contenía el proyecto base de Laravue, comentado en el apartado de Tecnología, y lo puse en la misma carpeta donde había creado el proyecto de Laravel. El problema está en que el proyecto de Laravue se ejecuta de la misma forma que el proyecto de Laravel, por tanto cuando intento ejecutar el proyecto de Laravel siempre ejecutaba el de Laravue. Decidí borrar la carpeta entera de Laravue del proyecto Laravel para ver si así se solucionaba el problema, pero continuaba abriendo el proyecto de Laravue, porque se ve que se modificó el archivo *package.json* del proyecto de Laravel al instalar alguna dependencia del Laravue y no sabía exactamente qué es lo que se había modificado.

9.2.1 Solución

Borré por completo la rama la cual estaba utilizando antes de clonar el proyecto de Laravue. Como solución al problema anterior he decidido separar los dos proyectos en dos carpetas diferentes, quedando de la siguiente forma:

Figura 9.5: Estructuración de carpetas en Github

jma86 Merge pull request #114 from jma86/iss111 ...			Latest commit 046388c 4 days ago
laravue	Proves en Laravue + reunion amb Jaume		28 days ago
lletraferit	Modificada l'opcio de editorials electroniques per obres		4 days ago
README.md	Initial commit		4 months ago

Elaboración propia

Así ya puedo ejecutar uno y otro por separado sin que siempre se abra uno solo.

9.3 Datos que no relacionan a ninguna tabla

A lo largo del desarrollo del proyecto, sobre todo en la parte del *front office*, a medida que iba mostrando los datos por pantalla, me he dado cuenta que hay algunos datos que están guardados en base de datos con un id: X pero que en realidad si observo la tabla a la cual relaciona no existe el id asociado. Algunos de los ejemplos que yo he ido apuntando en una lista de errores son los siguientes:

- Tabla *epublicacio*: aparecen editores como por ejemplo con id=13, el cual no existe en la tabla *lmeditors*.
- Tabla *epublicacio*: aparecen obras como por ejemplo con id=352, la cual no existe en la tabla *eobres*.
- Tabla *mmobrapremi*: aparecen obras como por ejemplo con id=13, la cual no existe en la tabla *eobres*.
- Tabla *mmautorobra*: aparecen autores como por ejemplo con id=77, el cual no existe en la tabla *eauteurs* a la hora de obtener las obras colaborativas (realizadas por varios autores).
- Tabla *eauteurs*: el campo *formacio* presenta algunos errores tipográficos, ya que los datos que presenta, aparecen en formato `<i>` `</i>` citando las diferentes formaciones de cada autor. En algunos casos, falta el cierre de la `</i>` por lo que toda la página aparece en cursiva.

9.3.1 Solución

En primer lugar, comentar que yo todos estos errores que he comentado anteriormente están reflejados en un listado de errores que se lo facilitare a mi cliente cuando tenga accesibilidad al *back office*. Lo que he hecho para solucionar todos estos problemas ha sido bloquear desde el servidor la posibilidad de que un usuario pueda consultar información que no existe en base de datos. Por ejemplo, en el caso anterior de que las obras no existían, el usuario en la web verá lo siguiente:

Figura 9.6: Vista para el usuario cuando hay errores en las relaciones de la base de datos



Elaboración propia

Para el resto de casos, he aplicado el mismo caso que he comentado. Estos errores una vez se eliminen o modifiquen en la base de datos ya no aparecerán, pero no está de más comprobar siempre si el dato existe por si el administrador del *back office* tuviera algún error, por tanto siempre se realizarán las comprobaciones y así si el usuario ve algo raro puede contactar con el administrador para comentarle la incidencia.

Capítulo 10

Conclusión y trabajos futuros

En este trabajo de final de grado, he conseguido implementar un proyecto web el cual terminará siendo una web real utilizada por:

- Escritores de literatura valenciana para que puedan tener accesibles desde internet todas sus obras y realizar cambios sobre ellas.
- Usuarios puedan consultar cualquier tipo de obras de varios géneros y subgéneros.
- Editoriales para que se puedan promocionar.

He realizado un típico caso de modernización de tecnologías a partir de una serie de sistemas antiguos. En este caso, he modernizado el antiguo sitio web de literatura valenciana, que pensando de cara a la vida profesional me puedo encontrar y enfrentar en numerosas ocasiones al mismo problema. Hoy en día, si no realizas una mejora continua, a la larga tiendes a desaparecer porque el sistema se queda obsoleto, sin posibilidad de funcionar y ofrecer un servicio a los usuarios. Es por ello, que siempre hay que actualizarse con las nuevas tecnologías para poder seguir ofreciendo mayor o igual calidad a nuestros usuarios. En mi opinión, estoy muy contento con el trabajo realizado, he aprendido un montón de cosas que pienso que de cara al mundo laboral me van ayudar un montón, sobre todo el tema de Laravue me ha parecido muy interesante y complejo.

Entre las cosas que he aprendido sobre todo de Laravel gracias a mi tutor han sido:

- [2] Que es un *Helper*, cómo se crea y utiliza en código PHP.
- El ciclo de vida de una petición AJAX: La utilización de Ajax para llamar a un *Controller* y cargar parte de la página al crear un evento.
- Algunas técnicas de optimizar el trabajo con el ORM de Laravel: *eloquent*.

Como continuación de este trabajo de final de grado, existen diversos puntos de trabajo que quedan abiertos y en los que es posible continuar trabajando en un futuro. Estos puntos se pueden clasificar en: funcionalidades no implementadas y ampliaciones de funcionalidad. A continuación se presentan algunos puntos de trabajo que por exceder el alcance del proyecto, no han podido ser tratados con la suficiente profundidad.

10.1 Funcionalidades no implementadas

- Completar la funcionalidad del formulario de “Actualització de dades”. El usuario podrá enviar un correo al administrador comunicando su incidencia.
- Implementar todos los CRUD de las tablas restantes en el *back office*:
 - Capitols
 - Comarca
 - Comarcanova

- Poblacio
- Poblacionsova
- Professions
- Subgenere
- CMS (Páginas estáticas)
- Autor-Critica
- Autor-Premi
- Colaboracions
- Obra-Idioma
- Obra-Premi
- Subscriptors

10.2 Ampliaciones de funcionalidad

- Enviar desde el *back office* correos a todos aquellos usuarios que estén suscritos a la web. Los correos contendrán contenido de interés para el usuario como: cambios realizados, nuevos libros, el libro del mes...
- Permitir que cada obra pueda tener más de una imagen.
- Nuevas prestaciones propuestas por Biel como el libro del mes.
- Reestructurar las rutas y controladores de la parte del *front office*:
 - Fusionar algunos *controllers* que son lo mismo.
 - Simplificar rutas.

10.3 Pruebas por realizar

Por último, comentar que me hubiera gustado poder realizar pruebas de carga/estrés/*random* data para comprobar lo siguiente:

- Nivel de fiabilidad del sitio web.
- Solidez en momentos de carga extrema -> determina si el sitio web rendirá lo suficiente en caso de que la carga real supere la esperada.
- Comportamiento del sitio web bajo una cantidad de peticiones esperadas.
- Tiempos de respuesta de transacciones importantes.

Capítulo 11

Referencias

- [1] Guía de uso de Laravue: <https://doc.laravue.dev/guide/#features>
- [2] Guía de uso de *Helpers* de Laravel: <https://laravel.com/docs/5.8/helpers>
- [3] Guía de uso de Bootstrap Vue: <https://bootstrap-vue.js.org/docs/components/>
- [4] Guía de uso de Bootstrap: <https://getbootstrap.com/docs/4.4/getting-started/introduction/>
- [5] Guía de uso de Bullzip: <http://www.bullzip.com/products/a2m/info.php>
- [6] Sitio web online de creación de *mockups*: <https://app.moqups.com/login>
- [7] Repositorio Github con el proyecto base de Laravue: <https://github.com/tuandm/laravue>
- [8] Sitio web para la plantilla del *front office*: <https://colorlib.com/wp/template/callie/>

